University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations Theses, Dissertations, and Major Papers

2000

Modeling of cellular manufacturing systems with productivity consideration: A simulated annealing algorithm.

Abduelghani I. Abduelmola
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

Abduelmola, Abduelghani I., "Modeling of cellular manufacturing systems with productivity consideration: A simulated annealing algorithm." (2000). *Electronic Theses and Dissertations*. 2753.
https://scholar.uwindsor.ca/etd/2753

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

MODELING OF CELLULAR MANUFACTURING SYSTEMS WITH PRODUCTIVITY CONSIDERATION: A SIMULATED ANNEALING ALGORITHM

by
Abduelghani I. Abduelmola

A Dissertation
Submitted to the College of Graduate Studies and Research
Through Industrial & Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

2000

α 2000 Abduelghani Abduelmola

National Library of Canada

Acquisitions and Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada

Acquisitions et services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-52416-7

Canada

ABSTRACT

Cellular Manufacturing (CM) is a manufacturing system in which similar parts and their required machines are grouped into manufacturing cells. The implementation of CM systems leads to increased output, decreased setup time, reduced work-in-process, reduced material handling cost as well as improved system productivity. One problem in the design of CM systems is cell formation (CF). Solving the CF problem in CM systems may lead to the organization or re-organization of manufacturing systems into manufacturing cells and to the determination of the type and number of machines required in each manufacturing cell.

Many models have been developed to solve the CF problem over the last two decades. A thorough literature review reveals that most models use an indirect index such as a similarity or dissimilarity index as an objective function. The use of an indirect measure may not reflect the true state of CM systems.

In this thesis, a CM system design model that uses a direct measure "productivity index" is presented. First, a 0-1 integer-programming model that maximizes the ratio of the output to the total material handling cost is developed to form part families and machine groups simultaneously. Second, a simulated annealing (SA) algorithm is developed to solve large-scale problems. This algorithm provides several advantages over some of the existing algorithms. It forms part families and machine groups simultaneously and considers production volume, selling price, and maximum number of machines in each cell. Moreover, it has the ability to determine the optimum number of manufacturing cells; so there is no need to specify the number of manufacturing cells in

advance. Several problems selected from the literature are used to test the performance of the developed models. The results showed the superiority of the SA algorithm over the integer-programming model in both productivity and computational time.

Furthermore, the majority of the existing models assume that each part has a unique process plan. In real manufacturing systems, however, a part can be produced using different routes and machines, which improves the productivity of CM systems. Hence, the developed models are extended to consider alternative process plans.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and thanks to Dr. S. Taboun, my advisor, for suggesting the topic, for his continuous help in its development, and for being a constant source of encouragement. This dissertation would never have been completed without his guidance and cooperation. I appreciate his time, effort, and encouragement.

My special thanks to Dr. S. Sankaran of the University of Regina for serving as the external examiner to the dissertation committee.

I would also like to express my sincere thanks to my committee members (in no particular order) Dr. D. Kao, Dr. M. Wang, and Dr F. Salustri for their time, useful comments and helpful advice in improving the quality of work during the course of my program. My thanks to Mr. R. Barakat, the Technician of the Industrial and Manufacturing Systems Engineering Department and Mr. R. Mac Isaac, the Corporate Forward Manger of the AG Simpson Automotive Systems for their unlimited help.

I am also indebted to my parents, brothers, and sisters who provided encouragement and support throughout my study. My special thanks to my wife for her great support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Indices

(i) Parts

$i = 1, 2, \ldots, p$

(ii) Machines

$j = 1, \ldots, m$

(iii) Process plans

$r = 1, \ldots, R$

(iv) Cells

$k = 1, \ldots, c$

Parameters:

b_{ij}	= 1	if part i needs machine j
	= 0	Otherwise
D_i	= annual demand of part type i	
E_{ir}	= 1	if part i uses process plan r
	= 0	Otherwise
EMC	= inter-cell handling cost	
I	= input	
IMC	= intra-cell handling cost	
M_{max}	= maximum number of machines in each cell	
NM_{ir}	= number of machines required by part type i using process plan r	

O = output

S_i = selling price of product i

Variables:

X_{jk} = 1 if machine type j is used in cell k

= 0 Otherwise

Y_{ik} = 1 if part i belongs to cell k

= 0 Otherwise

Y_{irk} = 1 if part i using process plan r belongs to cell k

= 0 Otherwise

GLOSSARY

Bottleneck Machine

A machine that is required by different part families from different manufacturing cells. This machine does not allow some parts to be manufactured in a single cell.

Exceptional Part

A part requiring processing in more than one manufacturing cell.

Inter-Cell Moves

The number of moves between cells needed by a part to complete its required operations.

Intra-Cell Moves

The number of moves between machines required by a part to complete its operations within a cell.

Machine Cell

A group of machines that are capable of producing some or all of the operations required by one or more part families.

Part Family

A group of parts that have similar operations to be processed in one or more manufacturing cells.

Partial Productivity

The ratio of output to any single input.

Process Plan

A manufacturing direction that determines which machining processes (cutting, welding, machining, inspection, painting, final assembly, etc) and parameters (setup time, processing time, tool, etc) are to be used to convert a part from initial form (input) to a final form (output).

Total Factor Productivity

The ratio of output to the sum of labour and capital input factors.

Total Productivity

The ratio of output to all inputs.

Void Part

A part that is assigned to a cell but does not require processing throughout all machines in that cell.

CHAPTER 1

INTRODUCTION

1.1 An Overview

Conventional manufacturing systems are classified into job shop, batch, and mass manufacturing systems (Figure 1.1). Job shop systems are well suited for high part variety and low volume while mass production systems are appropriate for high volume and low part variety. Batch manufacturing systems are appropriate for medium volume and variety.

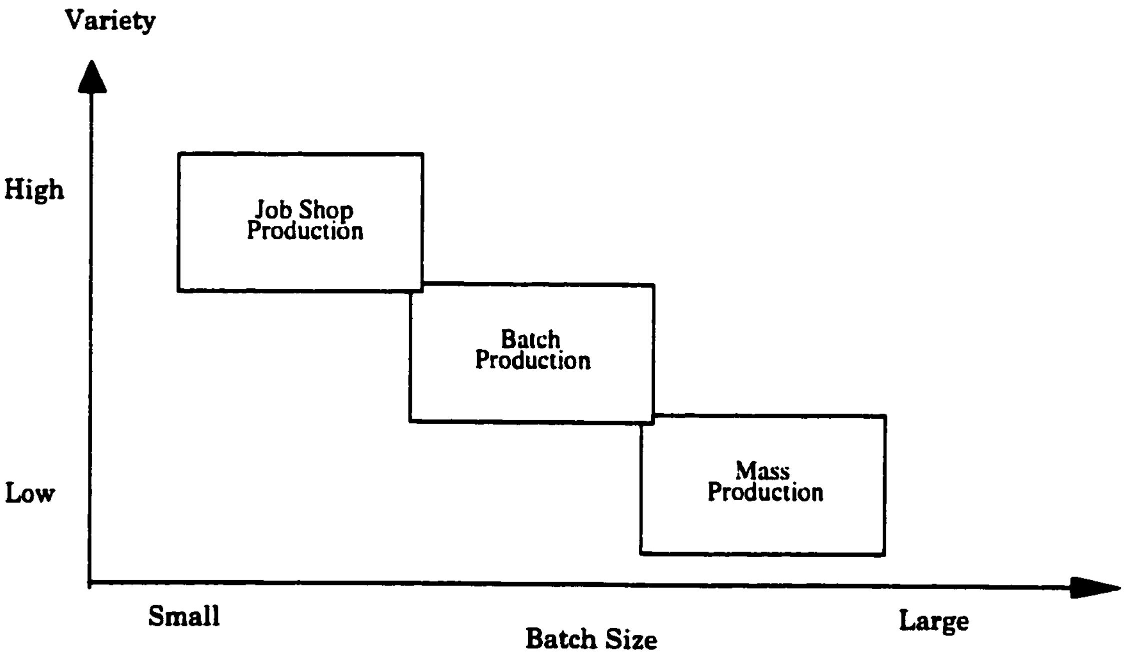


Figure 1.1. Different types of traditional manufacturing systems (Brown *et al.*, 1988).

Although traditional manufacturing systems are able to produce a high variety of parts in small batch sizes, producing small batch sizes has some limitations such as loss of production time due to increase in set up time, large number of movements, and large in-process inventories. Manufacturing a part requires about 5% of the processing time on machining, while the remaining 95% is spent in waiting and moving. Of the time spent on machining, less than 30% is spent on production. The remaining 70% is spent in machine loading and unloading (Figure 1.2).

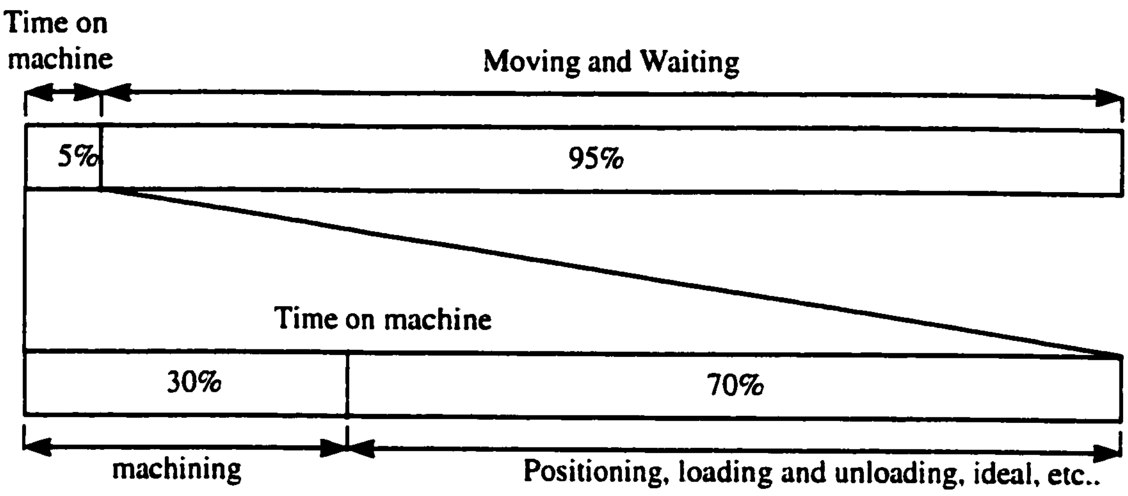


Figure 1.2. Manufacturing life cycle of a workpiece in a batch system (Ham *et al.*, 1985).

In traditional manufacturing systems, machines are grouped based on their types; for example, drilling machines are grouped in a single cell, and broaching machines in another cell as shown in Figure 1.3. Parts that require different machines have to be routed throughout different manufacturing cells to complete the required operations. This extensive movement increases total material handling cost and decreases system productivity. These limitations are forcing traditional manufacturers to consider changing and improving their facilities to improve productivity. Cellular manufacturing (CM) systems have been implemented for traditional manufacturing systems and have greatly minimized these limitations (Knauss and Matuszak, 1989; Burbidge and Halsall, 1994).

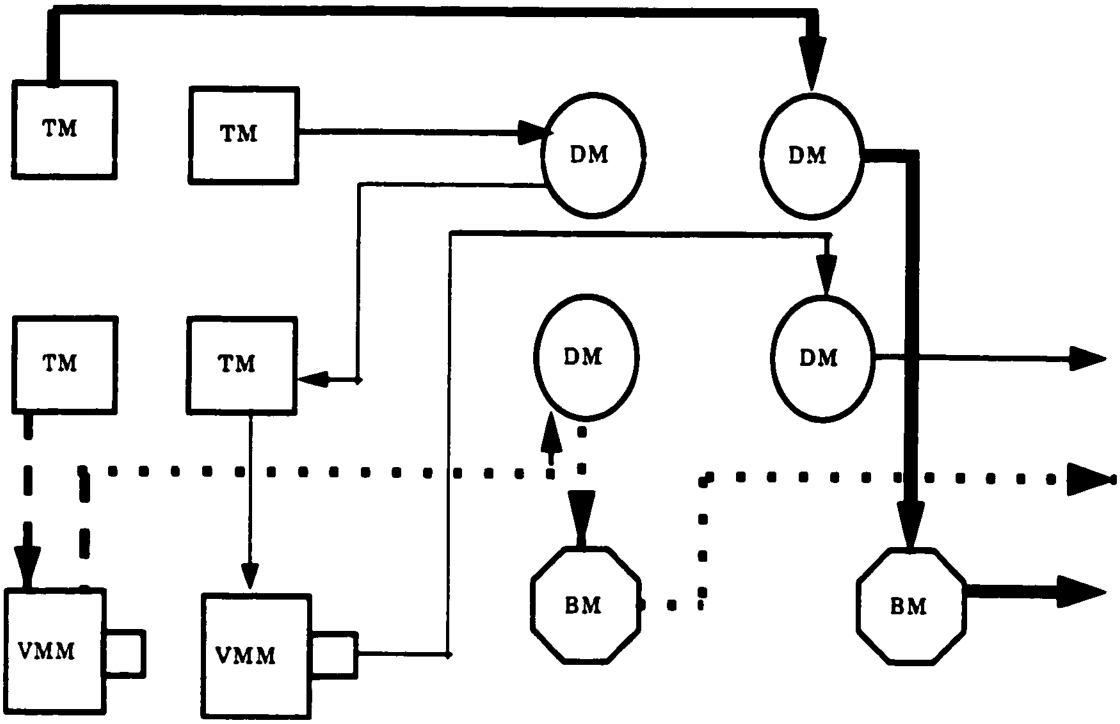


Figure 1.3. Arrangement of cells in a job shop production system (Heragu, 1994).

Note: BM:Broaching Machine; DM:Drilling Machine; TM:Turning Machine; VMM:Vertical Milling Machine.

1.2 Cellular Manufacturing Systems

Cellular manufacturing is an application of group technology (GT) in which manufacturing systems are re-arranged into manufacturing cells by grouping similar parts into part families and their required machines into manufacturing cells. In CM systems, machines are grouped based on the process planning of parts (Figure 1.4). Clearly, parts require fewer movements to complete the required operations in CM systems than in traditional manufacturing systems.

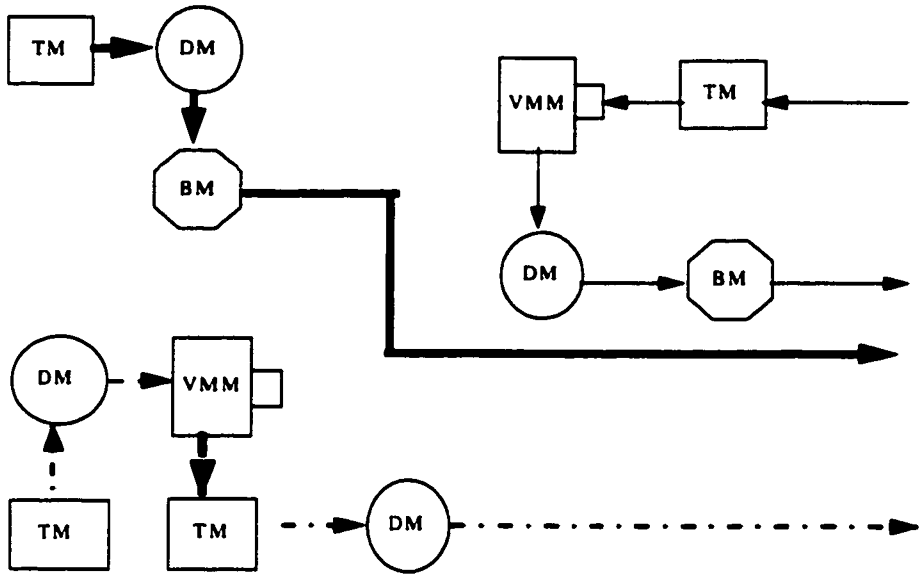


Figure 1.4. Arrangement of cells in a CM system (Heragu, 1994).

Note: BM:Broaching Machine; DM:Drilling Machine; TM:Turning Machine; VMM:Vertical Milling Machine

Applying CM to traditional production systems can minimize some of their disadvantages. Studies have showed that manufacturing companies that had implemented CM systems improved their productivity level by gaining the following benefits (Burbidge & Halsall, 1994):

- Increased output
- Increased rate of return on investment
- Increased profit
- Improved quality

- Decreased setup time
- Decreased overdue orders
- Decreased work in-process inventories
- Reduced material handling cost
- Reduced space requirements
- Simplified scheduling

Several design and operational problems should be addressed in order to implement CM systems successfully. Design problems include identifying part families and machine groups, as well as selecting process plans, material handling equipment, tools, and fixtures. Operational problems related to managing and operating manufacturing systems include scheduling, inspection, maintenance, and control. The first problem in CM systems design is the Cell Formation (CF) or grouping problem since most decisions are based on it. This thesis focuses on the CF or grouping problem.

1.3 The Cell Formation Problem

Solving the cell formation problem in CM systems leads to the organization or re-organization of manufacturing systems into manufacturing cells and to the determination of the type and number of machines required in each cell. Different approaches have been developed to solve the cell formation problem. One approach involves classification and coding of parts based on their features such as diameter, material, and so forth. Many classification and coding systems have been developed to form part families. Ham *et al.* (1985) and Joines *et al.* (1996) presented a comprehensive survey of these systems. These codes help to minimize the unnecessary variety of components by making the designer aware of similar parts. However, this approach is expensive and needs considerable effort to design and implement (Kusiak 1987).

Because of these limitations, many techniques have been developed to solve the cell formation problem based on the production flow analysis (PFA) introduced by Burbidge (1963). In PFA, the relationship between machines and parts is represented by a 0-1 machine-part matrix (part routing sequence) as shown in Table 1.1. In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*. Rearranging the rows and columns in the part-machine matrix (Table 1.1) results in two part families PF1 ={1, 3} and PF2 ={2, 4, 5, 6} and two machine groups MG1 {1, 4, 6} and MG2 {2, 3, 5, 7} as shown in Table 1.2.

Several applications and survey papers on techniques for solving the CF problem have been published (King and Nakornchai, 1982; Mosier, 1985; Chu, 1989; Taboun *et al.*, 1991; Kusiak, 1992; Sankaran and Kasilingam, 1993; Heragu, 1994; Liagn and Taboun, 1995; Joines *et al.*, 1996; Abduelmula *et al.*, 1997; Selim *et al.*, 1998; Abduelmula *et al.*, 1998; Taboun *et al.*, 1998a and 1998b). These techniques include matrix arrangement methods, similarity coefficient methods, graph theoretical methods, mathematical programming methods, and heuristic methods. Most of these methods determine the optimal solution; however, they have several drawbacks. First, grouping of parts and machines is always achieved by maximizing or minimizing an indirect index such as a similarity or dissimilarity index. Second, most of these techniques operate only on the machine-part matrix and do not consider important factors such as production volume, and inter-cell cost. Third, they do not simultaneously form part families and machine groups. Fourth, they require visual inspection by the user to identify the final manufacturing cells. Last, they require long computational time to solve large-scale problems. Therefore, a CM design model that considers the total material handling cost and overcomes those drawbacks is presented in this thesis.

Part	Machines						
	M1	M2	M3	M4	M5	M6	M7
P1	1	0	0	1	0	1	0
P2	0	1	0	1	1	0	0
P3	0	0	0	1	0	1	0
P4	0	1	1	0	0	0	0
P5	0	1	1	0	0	0	1
P6	0	1	0	0	1	0	1

Table 1.1. Binary part-machine Matrix.

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*. Each part has one process plan.

Part	Machines						
	M1	M4	M6	M2	M3	M5	M7
P1	1	1	1	0	0	0	0
P3	0	1	1	0	0	0	0
P2	0	1	0	1	0	1	0
P4	0	0	0	1	1	0	0
P5	0	0	0	1	1	0	1
P6	0	0	0	1	0	1	1

Table 1.2. Part families and machine cells.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

1.4 Material Handling

The main objective of material handling systems is to deliver a workpiece or finished part to the appropriate workstations at the right time. Material handling activities include receiving a workpiece from suppliers, moving the workpiece from receiving to the workstations, moving the workpiece among the workstations, and moving the finished part to the final storage area. Increased material movement increases total material handling cost and decreases system productivity.

The main objective of CM systems is the formation of manufacturing cells with no inter-cell movement; that is, each part family is completely produced within one manufacturing cell. However, due to high investment and maintenance costs of some machines, and due to technological constraints such as space limitation, and safety, the production of some part families may require more than one manufacturing cell. It is important, therefore, to consider both intra and inter-cell material handling costs while designing CM systems and to eliminate inter-cell movement if possible.

The intra and inter-cell movements have been described in terms of number of voids and exceptional elements respectively (Adil *et al.*, 1996; Srinivasan and Narendran, 1991). However, both researchers do not consider important factors such as production volume, selling price, and maximum number of machines in each cell due to space limitation. Increasing the number of voids leads to large cells, which in turn increases the number of movements within a single cell. Higher intra-cell movements increase the intra-cell handling cost. Furthermore, the introduction of exceptional parts increases the number of parts that require machine resources located in multiple cells. This increases the inter-cell movement cost.

One way to decrease the number of intra and inter-cell movements is to consider alternative process plans. For example, suppose that part 2 in Table 1.1 has an alternative process plan that involves machines 2, 3, and 5. This will reduce the number of voids from 7 to 6 and the number of inter-cell movements from 1 to 0. Therefore, it is important to consider different process plans when designing CM systems.

1.5 Organization of the Thesis

The following chapter presents a comprehensive literature review of the existing cell formation techniques as well as productivity measurement and optimization models. Chapter 3 presents the design of a CM system considering total material handling cost. A simulated annealing algorithm is presented in chapter 4. CM design considering alternative process plans is presented in chapter 5. Conclusions and directions for future research are presented in chapter 6.

CHAPTER 2

LITERATURE REVIEW

Over the last two decades many researchers have aggressively studied the CF problem in CM design as well as productivity measurement and optimization. Many models have been developed to solve the cell formation problem. At the same time there is a growing interest in defining productivity and in studying the relationship between input and output factors. In this thesis, the literature review is divided into two parts: cell formation and productivity measurement and optimization.

2.2 Cell Formation Problem

Many researchers have addressed the cell formation problem (King and Nakornchai, 1982; Mosier, 1985; Chu, 1989; Kusiak, 1992; Heragu, 1994; Joines *et al.*, 1996; Selim *et al.*, 1998; Taboun *et al.*, 1998). In this thesis, methods for solving the cell formation problem are classified as shown in Figure 2.1 and are briefly reviewed in the following sections.

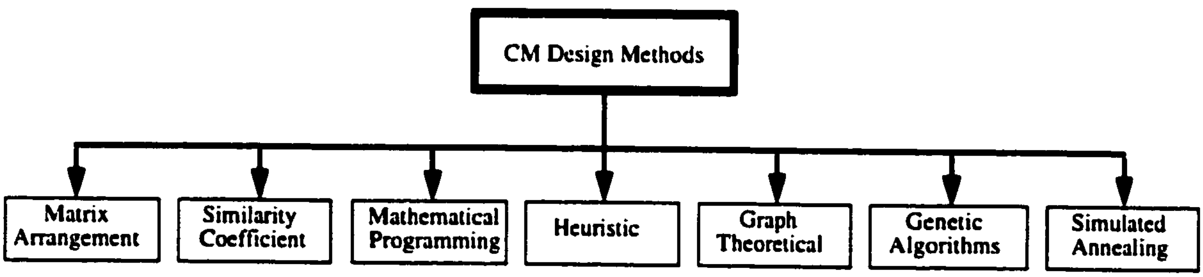


Figure 2.1. CM systems design methods.

2.2.1 Matrix-Arrangement Methods

These methods deal with rearrangement of rows and columns of a part-machine matrix to form a block diagonal matrix. Each block represents a manufacturing cell. In these methods, part families and machine groups can be formed simultaneously. However, they do not include technological factors such as machine capacity, processing time, production volume, etc., and therefore the solution may not reflect the real state of manufacturing facilities. Some of these methods are briefly reviewed in the following paragraphs.

King (1980) developed an algorithm to solve the cell formation problem called the rank order clustering (ROC) method. The block diagonal form can be obtained if the machine-part processing matrix is well structured, that is the matrix has few exceptional elements. The main advantage of this method lies in its ability to deal easily with the problems of bottleneck machines and exceptional elements. However, the accuracy of results is dependent on the original machine-part matrix.

Chan and Milner (1982) developed an algorithm known as the direct clustering algorithm. The algorithm rearranges the machine-part matrix until there is no more improvement due to restructuring. This algorithm allows users to deal with the problem of bottleneck machines and exceptional elements.

Kusiak (1987) introduced a 0-1 integer linear programming formulation known as the p-median model to group parts into part families. This model aims to maximize the total sum of similarity coefficients between two parts for a fixed number of cells. This model assumes that each part has only one process plan, and is assigned to one cell only. However, the number of cells has to be specified in advance. Kusiak also presented a new model named the generalized P-median model, which allows different process plans.

Amira and Choobineh (1996) presented a two-stage procedure for identifying the manufacturing cells in CM systems. The first stage presents an algorithm for identifying a pure block diagonal structure as well as bottleneck machines and/or exceptional parts. The second stage presents a 0-1 integer-programming model. This model aims to minimize the subcontractor and inter-cell movement costs, and the duplication cost of bottleneck machines. Implementation of the second stage depends upon the results of the first stage, that is the second stage is necessary if the first stage produces exceptional parts and/or bottleneck machines. A numerical example is presented to illustrate the procedure of the second stage.

Veeramani and Main (1996) proposed a two-stage algorithm to solve the cell formation problem. In the first stage, the matrix that includes exceptional elements is chosen from the incident matrix and the resulting disjoint sub-matrices of the matrix are identified. In the second stage, the column of the chosen matrix is reassigned to the sub-matrices such that the number of exceptional elements is minimized.

2.2.2 Similarity Coefficient Methods

These methods identify part families and machine groups based on the similarities between parts or machines. Similarities are based on machines, tools, and fixtures required by parts. This method cannot group part families and machine groups simultaneously and needs an additional procedure to solve the problem completely. Many measures of similarity among parts and machines have been developed to form part families and machine groups. Some of these measures are briefly described in the following paragraphs.

McAuley (1972) first applied the similarity coefficient to solve the cell formation problem. The similarity coefficient between two machines was defined as the ratio of the number of parts that visit both machines to the number of parts that visit at least one of these two machines. He introduced a 0-1 matrix, constructed with rows representing machines and columns representing parts. He also developed a new technique known as the single-linkage clustering technique. In this technique, a new machine is added to an existing cell if the similarity coefficient between the new machine and any existing machine in the cell exceeds a certain level.

Carrie (1973) developed a technique that identifies part families using a similarity coefficient between parts. In this model, the similarity coefficient between parts is determined by the ratio of the number of machines visited by both parts to the number of machines visited by one of the parts. A new constraint that specifies the maximum number of parts in each cell is also added.

Waghodekar and Sahu (1984) developed a method known as machine-component cell formation (MACE). In this method, three similarity measures have been used. The similarity coefficient proposed by Carrie (1973) is used to determine the group of machines. A similarity coefficient based on total number of parts is used to determine the inter-cell movement and manufacturing cells. A Similarity coefficient based on total flow of common parts processed by a particular machine type. A practical problem is used to illustrate the proposed method. The method yields a minimum number of exceptional elements. It is computationally straight and easy to understand.

Choobinh (1988) developed a procedure for solving the cell formation problem in two stages. In the first stage, clustering techniques with a similarity measure using the manufacturing operations and operation sequences are used for forming the part families. In the second stage, an integer-programming model is developed to form the machine cells and to assign the part families to the cells. The objective function aims to minimize the production, purchasing, and maintenance costs of machines. This model assumes that an operation can be performed on more than one machine.

Wei and Kern (1989) presented a new algorithm for solving the cell formation problem. This algorithm is based on the similarity between two machines. It considers the limitations on the size and number of cells. The algorithm is flexible in the sense that it can adapt to accommodate a wide variety of constraints relevant to the number and size of the cells.

Gupta and Sefoddini (1990) presented a similarity coefficient approach to solve the machine-part grouping problem. This measure uses the information for each operation performed on a part type in the early stage of the machine-part grouping procedure. The information includes production volume, processing requirements of parts, and unit production time. This approach ignores the cost of duplicating the bottleneck machines. Taboun *et al.* (1991) discussed the comparison and evaluation of three grouping criteria used to solve the cell formation problem. They selected grouping criteria based on measures related to machines required, processing sequence, and cutting tools required for production operations. A simulation model was also developed to illustrate the problem using SIMAN. They demonstrated that grouping obtained from machine similarity was superior for both evaluation criteria.

Kusiak and Cho (1992) developed two similarity coefficient methods. The first method was able to solve the cell formation problem considering alternative processing requirements. The second method generalized the first to be suitable for an ill structured matrix (a matrix that has some exceptional elements in its final form). Examples are also presented to illustrate the developed methods.

Kamrani and Parsaei (1993) presented a two-phase methodology for solving the cell formation problem. In the first phase, parts are grouped into part families according to their similarity in design and manufacturing features. In the second phase, machines are grouped based on relevant cost criteria: maintenance, material handling, and inspection costs.

Shafer and Rogers (1993) presented a new similarity measure for solving the cell formation problem using the similarity between two machines. The similarity between two machines (i, j) is computed as the maximum of two ratios in two different ways: the numbers of parts that visit both machines over the number of parts that require machine type i, and the number of parts that visit both machines over the number of parts that visit machine type j. The similarity between two parts is computed as the maximum of two ratios. In the first, it is computed as the total number of machines required by both parts over the number of machines required by the first part. In the second, it is computed as the number of machines that the two parts have in common over the number of machines required by the first one.

Suer and Ortega (1994) proposed a new similarity coefficient measure for forming part-machine cells. This measure incorporates both the types of machines and the number of machines required. However, it does not consider the sequence of operations or production volume of parts.

Viswanthan (1996) proposed a new similarity coefficient measure that aims to eliminate the main disadvantage of the p-median formulation developed by Kuiask (1987). The main advantage of this method is that it identifies the similarity and dissimilarity between machines. This method is tested using several examples taken from the literature, and the results show that the proposed method is more efficient than the existing methods.

Hwang and Ree (1996) suggested a two-stage procedure to solve the cell formation problem. In the first stage, a mathematical model is developed to solve the route selection problem. In this model, a part is allowed to have only one process plan. In the second stage, part families are formed based on the results obtained from the first stage using the p-median formulation. This model aims to maximize the similarity coefficient of parts in the same family. The machines are formed based on the minimum number of exceptional elements. A numerical example is presented to illustrate the solution procedure and the results are compared with those obtained from applying the p-median model. The grouping solution obtained by applying the proposed model is better than those obtained by applying p-median; however, it takes more computational time to determine the optimum solution.

Geonwook *et al.* (1998) presented a two-stage methodology to solve the cell formation problem in CM systems design. In the first stage, a similarity coefficient is presented to form part families considering alternative process plans. In the second stage, scheduling and operational aspects of the cell formation problem during machine failure are simultaneously considered. The objective of the proposed methodology is to minimize operating, inventory holding, and machine investment costs.

2.2.3 Mathematical Programming Methods

Mathematical programming methods show the functional relationship among variables through the use of mathematical symbols and operations. Mathematical programming methods are applied to the cell formation problem to find the optimal solution. In these methods, part families and machine groups can be formed simultaneously. They can model complex decision-making problems. Some of these models are presented in the following paragraphs.

Han and Ham (1986) developed a mathematical programming model to solve the CF problem based on the use of a classification and coding system. This model aims to minimize the distance function between any two parts. It forms mutually exclusive cells; i.e. each part can belong to only one cell. As this model can solve only part families, another procedure is needed to complete the solution.

Seifoddini (1989) developed a procedure to investigate the economic trade-off between machine duplication and inter-cell movement in the cell formation problem. This procedure was based on the decomposition of the machine-part matrix, and it considered some technological factors such as production volume and production time.

Gunasingh and Lashkari (1989) presented a sequential approach to solve the cell formation problem using 0-1 integer programming. In their model, the grouping of parts and machines into cells is based on the capabilities of the machines to process the parts. The approach starts by considering the machine groups based on their similarity in part processing. Then the allocation of parts to machine groups based on the processing requirements follows.

Rajamani *et al.* (1990) considered the CF problem, in which each part has alternative process plans and each operation can be performed on alternative machines. They developed three integer-programming models based on the availability of alternative process plans. The first model provides information to formulate a part-machine process matrix that can be solved using existing cell formation techniques. The second model forms cells assuming part families are known, while the third identifies part families and machine cells simultaneously.

Askin and Chiu (1990) proposed a mathematical model to solve the cell formation problem. Machine depreciation, inventory, material handling, and setup costs are first incorporated into a mathematical model. The cost model is then divided into two sub-problems to facilitate the solution. The first sub-problem determines the assignment of parts to machines, while the second sub-problem determines the assignment of machines to cells.

Jain *et al.* (1991) addressed the combined problem of cell formation and tools as a 0-1 integer-programming model. This model aims to minimize the overall system cost including processing, tools, and machines costs. The solution forms machine-part cells and decides on the number of machines and on the copies of tools required. In this model, the material handling cost is not considered.

Damodaran *et al.* (1992) developed a mixed integer-programming model that aimed at minimizing the costs of performing the operations, refixturing, and material handling. The solution of this model assigns part operations to machines. Examples illustrating the effect of material handling and refixturing costs on the operation allocation are presented.

Logendram (1993) developed an integer-programming model for solving the machine-part grouping problem. The objective function of the model focuses on maximizing a measure of effectiveness evaluated as the sum of the total number of moves and in-cell utilization. The model assumes that the desired number of manufacturing cells can be computed in advance by management personnel.

Dahel (1995) developed a 0-1 integer-programming model to determine machine groups and part families. The objective of this formulation is to minimize inter-cell movement for parts requiring processing in more than one cell. The model also considers part setup and processing times for the evaluation of capacity requirements.

Boctor (1996) developed a mixed integer-programming model to form part-machine cells. This formulation aims to minimize machine duplication and material handling costs. It allows the user to select the number of cells, control the cell sizes, and add extra machines if necessary. However, it does not consider other costs such as those for setup and labour.

Abduelmula *et al.* (1997) presented a methodology to optimize the total productivity of cellular manufacturing (CM) systems. The proposed methodology consists of two stages. In the first stage, an integer-programming model is developed to maximize the output. In the second stage, a 0-1 integer-programming model is developed to design a CM system. In this model, part families and machine cells are formed simultaneously. An example problem selected from the literature is used to illustrate the developed models.

Wang and Roze (1997) developed a non-linear integer-programming model modified from the P-median model developed by Kusiak (1987). This model aims at forming part families or machine groups. When used for forming part families, the model maximizes the similarity of parts subjected to a limitation on the maximum number of parts in each cell. It maximizes the machine similarity to form machine groups subjected to a limitation on the maximum number of machines per cell.

Abduelmula *et al.* (1998) proposed a two-stage CM productivity model. In the first stage, a 0-1 integer-programming model is developed to design CM systems. In this model, part families and machine cells are formed simultaneously. In the second stage, a total productivity model for CM systems is developed to optimize the total productivity of manufacturing cells by further using the unutilized machining time available on various machines.

Taboun *et al.* (1998b) developed an integer-programming model, which simultaneously forms part families and machine groups. The model aims at minimizing the various system costs including capital investment, part subcontractor, and material handling. A heuristic algorithm is also presented to solve large-scale problems. Different problems are used to test the developed models, and the results show that the proposed models are able to find an optimum solution and handle large-size problems.

2.2.4 Heuristic Methods

Heuristic methods can be defined as decision procedures or rules that guide the search process toward solving a problem. Heuristic methods are based on the actions selected by the user, i.e. good heuristic rules lead to good solutions and bad ones lead to bad solutions. Good heuristics algorithms can also reduce the computational time required to find an optimum solution. However, bad heuristics may take a long computational time.

All cell formation methods may be considered heuristic methods except mathematical programming methods. Heuristic methods use some features of the other methods such as similarity coefficient to help in forming part families and machine groups. Although other cell-formation methods aim at determining the optimum solution, they require long computational times to solve large-size problems. In these cases heuristic methods are used to find an optimum solution.

Steudel and Ballakur (1987) developed a similarity measure known as the Cell Bond Strength (CBS) which depends on part routing and production requirements. A two-stage algorithm was developed based on the developed CBS. The first stage forms the optimum group of machines so that the optimum bond among machines is maximized. The second stage partitions the group to form the machine cell.

Askin and Subramanian (1987) presented a heuristic approach for solving the machine-part grouping problem. This model considered machine, processing, setup, work-in-process, and intra-cellular material handling costs. An example problem was presented to test the algorithm.

Ballakur and Steudel (1987) developed a two-stage heuristic model to solve the cell formation problem. In their model machines are grouped into cells based on work load and cell-size constraints in the first stage. In the second stage, parts are assigned to cells so that a majority of their operations are performed within the cell. This model considers several constraints such as the maximum number of machines in each cell, work load restriction, and the number of parts produced by each cell. The algorithm is tested using several examples from the literature.

Harhalakis *et al.* (1990) proposed a two-stage heuristic algorithm to solve the cell formation problem. In the first stage, machines are grouped into cells based on the volume or cost of the material flow. The second stage aims at improving the first one by converting the inter-cell movement to intra-cell movement. The developed algorithm is applied to a large-scale problem.

Sule (1991) developed a procedure to determine the number of machines, their cells and the amount of material transfer between the cells. This model aims at minimizing the total production costs including machine capacity requirement cost, material handling cost, and machine fixed cost. The model does not allow for a subcontractor.

Okogba *et al.* (1992) developed an algorithm to solve the part-machine cell formation problem in three stages. In the first stage, machine groups are formed. In the second stage, machines are reallocated to minimize the number of inter-cell movements. Parts are assigned to associated machine groups that perform the maximum number of their operations in stage three. The developed algorithm is evaluated using a simulation model to investigate the performance of machine utilization and flow time.

Heragu and Gupta (1994) presented a heuristic method for forming part families and machine groups. This algorithm addresses several constraints such as machine capacity, technological requirements, cell size, and number of cells. It is assumed that the required number of machines and the material handling system have already been specified. A mathematical model is used to find the number of each machine type required by the parts. The algorithm then forms part-machine cells. The main advantage of this algorithm is that the user can change the values of the parameters and arrive at a new solution quickly.

Lin *et al.* (1996) proposed a two-stage integer-programming model for forming part-machine cells. The first stage determines an initial form of the machine-part incidence matrix into production cells. The second stage searches for improvements in the design of production cells. The proposed algorithm is compared with three other algorithms from the literature, and the results show its highly efficient computational performance. An industrial application is also presented.

Beaulieu *et al.* (1997) presented a two-phase heuristic method to solve the cell formation problem. The first phase aims to form manufacturing cells considering machine and intra-cell costs. It also considers alternative process plans. The second phase considers the introduction of inter-cell movement to improve machine utilization. An example is presented to test the proposed algorithm.

2.2.5 Graph Theoretical Methods

In these methods, the machine-part matrix is represented by a graph. Nodes represent machines and parts, and arcs represent the processing of parts. One set of nodes represents parts and the other set represents machines. The aim of these methods is to obtain sub-graphs from the machine-part graph to identify part families and machine groups. For example, Figure 2.2 presents a graph formulation for the machine-part matrix presented in Table 1.1. As shown in Figure 2.3, the graph formulation decomposes into two sub-graphs, with each sub-graph representing a part family (PF) and machine group (MG). The first sub-graph includes PF1 ={1, 3} and MG1 {1, 4, 6} while the second one includes PF2 ={2, 4, 5, 6} and MG2 {2, 3, 5, 7}. In graph theoretic methods, part families and machine groups are formed simultaneously. However, they are not suitable for large problems.

Rajagopalan and Battra (1975) presented a method for forming part families and machine cells in three phases. The first phase represents the production data and finds the similarity between machines. It then finds the cliques from the graph. They defined a clique as a sub-graph such that there is an edge or arc connecting every node pair in a sub-graph. The second phase uses the cliques as machine groups. Parts are assigned to machine cells in the third phase.

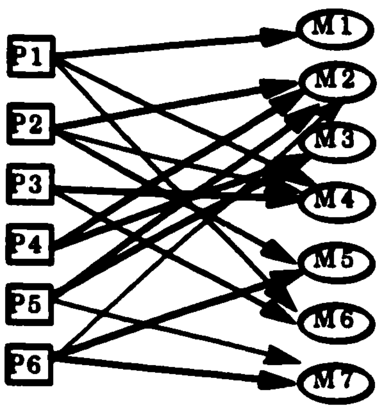


Figure 2.2. Graph formulation for the machine-part matrix presented in Table 1.1.

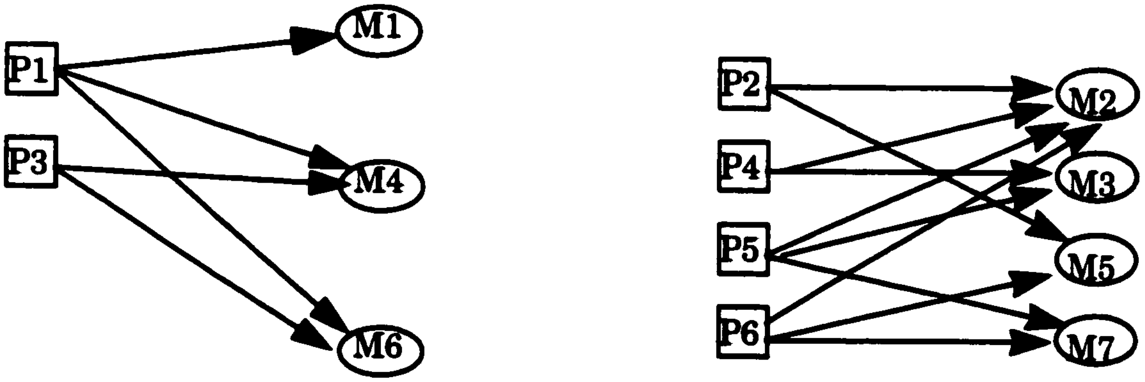


Figure 2.3. Two sub-graphs for the machine-part graph in figure 2.2.

Lee and Garcia-Diaz (1993) developed an approach to solve the cell formation problem. This approach aims at measuring the functional similarity between machines and then graphing machines into cells so all parts in each family can be processed in a machine cell.

Lee and Garcia-Diaz (1996) proposed a three-phase procedure to solve the machine-part grouping problem. The first phase determines the functional relationship between machines based on a machine-part matrix that specifies operation sequences and production quantity of each part. In the second phase, a 0-1 integer-programming model that seeks to minimize the material handling is developed. In the third phase, a 0-1 integer-programming model is proposed to form the part families. A numerical example is presented to show the solution procedure, and an industrial application is also presented.

2.2.6 Genetic Algorithms

A genetic algorithm (GA) is a search technique that starts with an initial set of random potential solutions, named the population, and uses an evaluation process to improve upon them. Venugopal and Narendran (1992a) proposed a mathematical model with a solution procedure based on GA for forming part-machine cells. This formulation considers two objectives: minimizing the volume of inter-cell moves and minimizing the total within cell load variation. The algorithm is applied to a cell formation problem selected from the literature, and the results show that the developed algorithm can be used for solving large-size problems.

Hwang and Sun (1996) presented a genetic algorithm to solve the part-machine grouping problem. This algorithm incorporates relevant production requirements such as production volume, processing time, and cell size. The first phase forms machine groups while the second phase assigns parts to the associated machine group. The proposed algorithm is compared with one of the existing algorithms. The developed algorithm is shown to be more effective in terms of flexibility and efficiency.

Joines *et al.* (1996) developed a genetic algorithm to solve the cell formation problem. This formulation allows the designer to remove the constraints on the number of allowable cells. The performance of the developed algorithm is compared with several traditional grouping methods. The results show that the developed algorithm is more flexible and effective than the other methods.

Escoto *et al.* (1998) presented a three-phase algorithm to form part families and machine groups. The first phase minimizes the dissimilarity of the part families. The second phase minimizes the number of operations needed to finish each part outside the part family. The third phase determines the load for each machine and assigns each machine to a machine group. An industrial application is presented to test the proposed algorithm.

2.2.7 Simulated Annealing

Simulated Annealing (SA) is a search method based on physical annealing of solids. Annealing is a process in which a solid is heated to liquid state followed by a slow cooling process. The aim of this process is to find the minimum energy level of the solid. It was first introduced by Metroplis *et al.* (1953) and applied by Kirkpatric *et al.* (1983). The algorithm begins by choosing an initial random feasible solution known as the current solution. Then inducing a small change in the current solution randomly generates a neighboring solution and the change in the objective function is calculated. Unlike in other search methods, in SA only one neighbor solution is generated at each transition and compared to the current solution, which may reduce the computational time.

Venugopal and Narendran (1992b) presented a simulated annealing algorithm to solve the machine-part grouping problem. The developed algorithm is tested using a numerical example selected from the literature, and the results show the capability of the algorithm to solve large-size problems. This algorithm is easy to understand and implement.

Chen and Srivastava (1994) developed a simulated annealing algorithm for forming part families and machine groups. This model aims to maximize the sum of machine similarities within a cell subject to cell size. An example problem selected from the literature is used to test the performance of the proposed model.

Boctor (1996) proposed a simulated annealing algorithm to form part-machine cells. The problem is formulated as a mixed integer-programming model. This formulation aims to minimize machine duplication cost and material handling cost. The model allows the user to select the number of cells and add extra machines if necessary. However, it does not consider other costs such as setup cost and labour cost.

Adil *et al.* (1996) developed a simulated annealing algorithm to solve part families and machine groups simultaneously. This formulation aims to minimize the sum of voids and exceptional parts. It assumes that each part is assigned to only one cell, and a limited number of machines is assigned to each cell. The model considers some technological factors including alternative routing and additional copies of machines. A numerical example is presented to illustrate the model.

2.3 Productivity Measurement and Optimization

As a result of trying to measure the effects of the technological advance on economic development, different productivity models have been developed. However, most of those models concentrated on partial labour productivity. These models are briefly described in the following paragraphs:

Kendric and Creamer (1969) defined productivity as the ratio of outputs to inputs. They also developed three indices of productivity measurement: total productivity, total factor productivity, and partial productivity. The total productivity index is defined as the ratio of all outputs to input. Total factor productivity is defined as the ratio of net output to the total factor input (labour and capital). Partial productivity is defined as the ratio of output to any single input. Later, Kendric (1972) pointed out that since it does not include other important cost elements such as material, labour, and capital, the total factor productivity index does not reflect a true state of a production system. He also discussed the benefits that can be obtained from using partial and total productivity measures.

Carig and Harris (1973) defined total productivity as the ratio of output to input. The output is defined as the sum of the total sales and interest obtained from financial sources. The input includes labour, capital, material, and any other input such as energy, taxes and transportation.

Sumanth (1979) developed a total productivity model in which total productivity was defined as the ratio of tangible outputs to tangible inputs. This model demonstrated how input and output factors are defined and measured.

May and Denny (1979) developed a total factor productivity model in which input was defined as a function of capital, labour, and material. They estimated all factors of the production function using data for a production period 1949-1970 of selected US manufacturing companies.

Husband and Ghobadain (1981) examined the applicability of two existing productivity models to a batch production company. The two models were tested using data for a production period 1975-1979. Husband and Ghobadain also explained how these results could be used to develop a productive productivity model.

Salvendy (1982) developed a productivity model that defined total productivity as the output divided by the total cost. However, he did not explain how the input factors were defined and measured.

Mohamed (1986) developed a total productivity model. In this model, the total productivity is defined as the ratio of output to input. The output is defined as the sum of total sales, services, and related activities. The input includes labour, capital, material, energy, and other expenses. This model was tested using production data from a metal fabricating company in Canada for the period 1971-1982. Mohamed pointed out that the total factor productivity of human and capital input factors showed better results when compared to the total productivity measures.

Rastogi and Monhaty (1994) developed four models to optimize the total productivity of a manufacturing company. The first model aims to maximize the over all growth by reducing the total production cost and increasing the total output. The second model aims to decrease the input cost with marginal reductions in total output. The third model aims to reduce the total input cost without reducing the total output. The last one aims to increase the output without spending extra input. These models were applied to real manufacturing data. The results show that the first strategy achieved the highest productivity value.

Kampersad (1996) developed a productivity model to optimize the total productivity of a robotics assembly system design. This model aims to select the best performance of a robotics assembly cell. This model defines input factors as the sum of labour, capital, material, and other input costs. The output is defined as the total of the production value. An example is presented to illustrate the developed model.

Abduelmola *et al.* (1997b) developed a productivity model that takes into consideration inherent inputs and secondary outputs. The inherent inputs refer to all inputs that have an indirect relation to the total cost and can be expressed in monetary value. These include labour skill, incentive and so forth, while secondary outputs refer to all outputs produced by a facility and are not considered as finished products such as scrap. Real data for the production period 1977-1990 of a manufacturing facility are used to test the presented model. The results are compared with those obtained by using an existing model selected from the literature. The results show the superiority of the developed model in both productivity values and capability of handling partial productivity measures.

2.4 The Limitations of Existing Models and Motivation for this Research

A comprehensive review of the literature regarding CM systems design approaches as well as productivity measurement and optimization models has been presented in this chapter. Although differences exist in the definition of objective functions and technological constraints, models in this field aim to group parts into part families and their required machines into machine groups. These approaches are able to form part families and machine groups; however, they have some limitations:

- A thorough review of literature fails to reveal a single model that uses a direct index such as a productivity index to design CM systems. Identifying part families and machine groups is usually achieved by optimizing an indirect index such as a similarity or dissimilarity index.
- The majority of the models do not take into account the cost associated with inter-cell movement.
- Many models neglect several important factors such as selling price, demand, maximum number of machines allowable in each cell, machine capacity, sequence of operations, safety requirements, and machine reliability.
- Some models cannot produce part families and machine groups simultaneously, and require manual inspection to identify part families and machine groups.
- The majority of models require long computational times to solve large-scale problems.
- The number of manufacturing cells to be used is specified in advance.

Furthermore, most of these models assumed that each part has only one process plan and each operation can be performed on only one machine. However, that is not always the case; in real manufacturing systems a part can be manufactured using different process plans and machines. Therefore, a productivity model that considers alternative process plans and overcomes some of the limitations associated with the existing models is needed.

2.5 Thesis Statement

The research conducted in this thesis is directed at developing a productivity model to identify part families and machine groups simultaneously. The thesis of this research states that a productivity index can be used as a direct measure to solve the CF problem in cellular manufacturing design, and that such a model will eliminate some of the limitations apparent in the existing models. This model helps designers to determine part families and machine groups simultaneously and selects the process plan such that the system productivity is optimized. A solution methodology of the model is also presented in this thesis. The solution takes into consideration production volume, selling price, maximum number of machines in each cell, and total material handling cost.

2.6 Objectives for the Proposed Research

As mentioned previously, many researchers have focused on the development of new methods to solve the cell formation problem in CM systems. The literature review revealed that most of those methods consider indirect measures as an objective function to form part families and machine groups, which may not reflect the true state of manufacturing systems. The main objectives of the proposed research are as follows:

1. Develop a model to design a CM system by using productivity index.

- First, a 0-1 integer-programming model is developed to maximize the system productivity defined as the ratio of output to the total material handling cost. This model takes into consideration the following :

* Intra and inter-cell handling costs
* Selling price
* Production volume
* Maximum number of machines in each cell

- Second, the developed model is validated using data from the literature.
- Third, a simulated annealing algorithm is developed to solve large-scale problems taking into account the above factors.
- Fourth, a detailed analysis is conducted on different problems to test the effect of various annealing parameters on the solution quality.
- Fifth, the developed simulated annealing algorithm is validated using the same data used to validate the mathematical programming model.
- Sixth, a performance comparison between the developed models is presented.

2. Address the above points considering alternative process plans.

CHAPTER 3

CELLULAR MANUFACTURING DESIGN CONSIDERING MATERIAL HANDLING

This Chapter presents a productivity model for solving the part-machine grouping problem in cellular manufacturing (CM) systems. A 0-1 integer-programming model is developed to identify machine groups and part families simultaneously. The objective is to maximize productivity defined as the ratio of output to the total material handling cost. In this model, part families and machine groups are formed simultaneously. The model has the ability to find the optimum number of manufacturing cells so there is no need to specify the number of manufacturing cells in advance. Eight problems of different size and complexity selected from the literature are used to test the performance of the developed models. The results show the ability of the developed model to find the optimum solution for all problems.

3.1 Problem Background

In traditional manufacturing systems, small batch sizes adversely affect many factors such as production time, in-process inventory, and material handling. The implementation of CM systems improves these factors as well as system productivity (Burbidge & Halsall 1994). One problem in the design of CM systems is cell formation (CF). Solving the cell formation problem in CM systems aims at organizing or re-organizing of manufacturing systems into manufacturing cells and determining the type and number of machines required in each cell. It also aims to form manufacturing cells with no inter-cell movement among the cells. That is, each part family is processed within one manufacturing cell. However, due to high investment and maintenance costs of some machines, along with technological factors such as safety requirements, it is not always possible to form manufacturing cells with no material handling among them. Some parts require processing in more than one manufacturing cell. Therefore, it is important to consider both intra and inter-cell material handling costs when designing CM systems and then to minimize the inter-cell movement among the cells if possible.

So far, many models for solving the CF problem have been developed (King and Nakornchai, 1982; Mosier, 1985; Chu, 1989; Taboun *et al.*, 1991; Kusiak, 1992; Sankaran and Kasilingam, 1993; Heragu, 1994; Liagn and Taboun, 1995; Joines *et al.*, 1996; Selim *et al.*, 1998; Abduelmula *et al.*, 1997; Abduelmula *et al.*, 1998; Taboun *et al.*, 1998a and 1998b). The majority of these models identify part families and machine groups by optimizing an indirect index such as a similarity or dissimilarity index. However, using an indirect measure may not reflect the true situation of the designed CM systems. This chapter presents a methodology to optimize the productivity of a CM system using a productivity index. A 0-1 integer-programming model is developed to maximize the ratio of output to the total material handling cost. The following section presents the problem statement. Problem formulation is presented in section 3.3. Section 3.4 explains the tradeoff between intra and inter-cell movements. Sample problems are presented in section 3.5. Sensitivity analysis is presented in section 3.6.

3.2 Problem Statement

A manufacturing system consisting of several machines is considered. There are a number of parts to be processed, and each part needs a certain number of operations. The objective is to optimize the system productivity by organizing these parts and machines into manufacturing cells.

3.3 Problem Formulation

3.3.1 Assumptions

1. It is assumed that production rate is known at the beginning of the planning period.
2. The system consists of a number of machine types that provide the required machining capacity to meet the demand.
3. Each type of part has a unique process plan and needs a specific number of machines.

3.3.2 Mathematical Model (MM)

In this section, a 0-1 integer-programming model is developed to form machine groups and part families simultaneously. The objective function of the model maximizes the system productivity defined as the ratio of output to the total material handling cost. Several options can be chosen by a decision-maker to optimize the productivity of manufacturing systems. Rastogi and Mohanty (1994) presented four strategies for productivity optimization of manufacturing facilities (Figure 4.1). They suggested that the best strategy is to maximize the ratio of output to input. In this thesis, productivity optimization is defined as the optimum ratio of the output to input. The output captures the production value of the number of units produced while the input captures the total material handling cost.

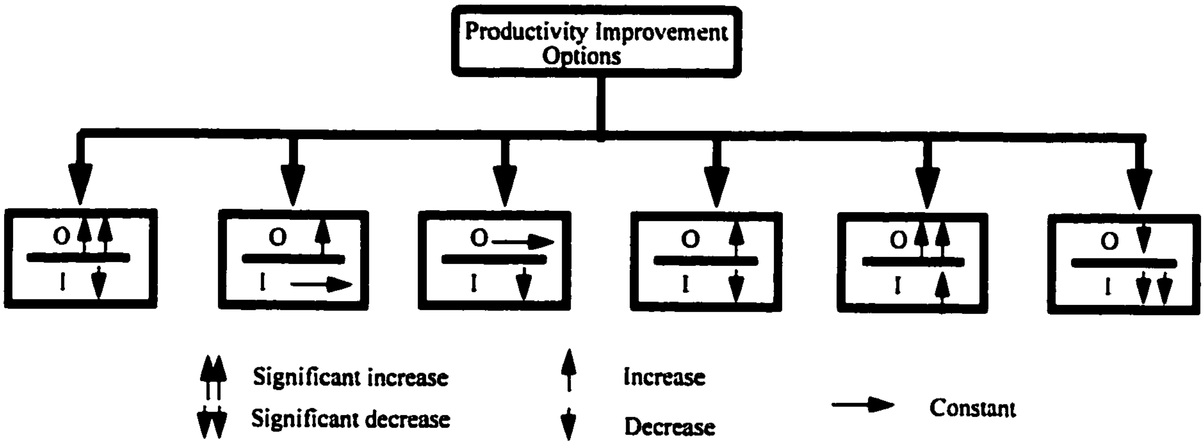


Figure 3.1. Productivity optimization options.

The model aims at maximizing productivity P, and can be formulated as:

Maximize

$$P = \frac{\sum_i \sum_k D_i * S_i * Y_{ik}}{\sum_i \sum_k NM_i * IMC * D_i * Y_{ik} + \sum_i \sum_j \sum_k (1 - X_{jk}) * b_{ij} * EMC * D_i * Y_{ik}} \quad (3.1)$$

Subject to

$$\sum_j X_{jk} < M_{max} \qquad \forall k \qquad (3.2)$$

$$\sum_k X_{jk} = 1 \qquad \forall j \qquad (3.3)$$

$$\sum_k Y_{ik} = 1 \qquad \forall i \qquad (3.4)$$

$$X_{jk} = 0 \text{ or } 1 \qquad \forall (j,k) \qquad (3.5)$$

$$Y_{ik} = 0 \text{ or } 1 \qquad \forall (i,k) \qquad (3.6)$$

The objective function of this model is presented in equation (3.1). The output is defined as the number of parts produced multiplied by their selling prices. The input captures the total material handling cost. The first term represents the intra-cell movement cost. The second term represents the inter-cell movement cost. The maximum number of machines in each cell is determined by constraint (3.2). Constraint (3.3) states that each machine is assigned to only one cell. Constraint (3.4) states that the final assignment of each part family is assigned to only one cell. Constraints (3.5) and (3.6) ensure the integrity of the decision variables.

3.3 Intra and Inter-Cell Movement Tradeoff

The tradeoff between intra and inter-cell costs is discussed by McAuley (1972) and Logendran (1990, 1991). Increasing the number of manufacturing cells increases the inter-cell cost, while the intra-cell cost decreases. Logendran (1990, 1991) stated that the appropriate minimum number of cells is 2 and the maximum number is dependent on some technological factors such as work force, space and budget constraints and is usually determined by management. The total material handling cost (TMHC) can be described as follows:

$$TMHC = X_1 n + X_2 m \quad (1)$$

Where:

X_1 = the intra-cell cost per movement

X_2 = the inter-cell cost per movement

n= the number of intra-cell movements

m = the number of inter-cell movements

In the literature, a higher value for inter-cell movement cost is assumed with respect to the intra cell cost. Logendran (1990, 1991) assumed a normal weight for θ_1 and θ_2 so that $\theta_1+\theta_2=1$. Where:

θ_1 = the weight placed on intra-cell cost per movement

θ_2 = the weight placed on inter-cell cost per movement

He assumed the values of 0.3 and 0.7 for θ_1 and θ_2 respectively, i.e.,

$$X_1 n = \theta_1 (TMHC) \tag{2}$$

$$X_2 m = \theta_2 (TMHC) \tag{3}$$

For the purpose of this research, the intra-cell cost is assumed to be \$2 per movement, i.e., X_1 n=2 (Kamrani & Parsaei, 1994). Since inter cell movement cost was not considered in their study, the same weights used by Logendran (1990, 1991) are used in this thesis. For n=1 and m=1, and by substituting in equation 2, we get TMHC=\$6.6, and therefore, X_2= \$4.6 per movement.

3.5 Illustrative Examples

In this section, 8 problems of different sizes and structures are considered to illustrate the developed model. Since information about demand and selling price were not included in all problems, a FORTRAN subroutine adapted from Microsoft FORTRAN is used to randomly generate that information (Appendix 1). Information regarding annual demand is generated between 200 and 300 units and between \$10 and \$15 for the selling price for all problems. For the purpose of this research, the intra and inter-cell movements are assumed to be \$2 and \$4.6 per movement respectively in all problems. Table 3.1 gives detailed information including the source from which these problems are obtained, the size of the problems and the maximum number of machines allowable in each cell.

All problems are solved using LINGO software using a Pentium 120 MHz computer. The input and output files for problem 5 are used as an example and presented in Appendix 2. The part-machine decision is made based on the values of Y_{ik} and X_{jk}. For example, Y_{610} indicates that part type 6 is assigned to machine cell 10, and X_{154} means that machine type 15 is assigned to group 4. The following paragraphs discuss the results obtained from applying the mathematical model to each of these problems.

Problem 1: This problem was considered by Mukhopadhyay *et al.* (1994). The part-machine matrix and the generated demand and selling price are presented in Tables 3.2 and 3.3 respectively. The optimum manufacturing cells of this problem is presented in Table 3.4. The problem was solved to optimality in 25 seconds. The solution has 2 manufacturing cells and an objective function value of 1.6953.

Problem 2: This problem is taken from Sarker and Balan (1996). Information regarding part-machine relationship is presented in Table 3.5. The randomly generated selling price and demand are presented in Table 3.6, while Table 3.7 presents the results obtained. The optimum number of manufacturing cells is 2, and the objective function value is 2.9733. The algorithm took 21 seconds to find the optimum solution.

Problem 3: This problem was considered by Mosier (1985). The original part-machine matrix is shown in Table 3.8, while demand and selling price are presented in Table 3.9. The solution obtained for this problem is presented in Table 3.10. The number of manufacturing cells is assumed to be 10 and the optimum number of cells obtained is 3. The optimum solution was found in 180 seconds with an objective function value of 1.9912.

Problem 4: In this problem, an 11 parts by 7 machines matrix (Table 3.11) used by Seifoddini and Djassemi (1996) is considered. The randomly generated demand and selling price are presented in Table 3.12. The maximum number of machines is assumed to be 4 and the number of cells is 7. Table 3.13 shows the final part families and machine cells. The solution results in 3 cells with an objective function value of 2.7584.

Problem 5: This problem was considered by Jayakrishnan and Narendran (1998). Information about part-machine relationships is presented in Table 3.14, while demand and selling price are presented in Table 3.15. The final part families and machine groups are summarized in Table 3.16. The maximum number of machines per cell is assumed to be 5 and the number of manufacturing cells is assumed to be 10. The problem is solved to optimality in 74 seconds. The optimum number of manufacturing cells is 4.

Problem 6. This problem was considered by Logendran (1990). The original part-machine matrix is shown in Table 3.17. Information about demand and selling price is presented in Table 3.18. The maximum number of machines is assumed to be 4, and the number of cells is assumed to be 7. Table 3.19 presents the final part families, machine cells and optimum number of cells. The model took 101 seconds and had an objective function value of 1.9156. The optimum number of manufacturing cells is 4.

Problem 7: A 20 parts by 8 machines problem (Table 3.20) presented by Chandrasekharan and Rajagopalan (1986) is considered. Information about demand and selling price is presented in Table 3.21. The number of manufacturing cells is assumed to be 8 and the maximum number of machines in each cell is assumed to be 4. Table 3.22

shows the final part families and machine groups as well as the optimum number of manufacturing cells. The model took 306 seconds to find the optimum solution with an objective function value of 1.2992.

Problem 8: A problem of 22 parts by 11 machines considered by Wen *et al.*, 1996 is selected. The part-machine relationship is presented in Table 3.22. Information about demand and selling price is presented in Table 3.23. The maximum number of machines per cell is assumed to be 4, and the number of cells is assumed to be 11. Table 3.24 presents the final part families and machine groups as well as the optimum number of cells. The optimum solution was obtained in 776 seconds with an objective function value of 0.7957.

Prob. No.	Source	Size (P*M)	M_{max}
P1	Mukhopadhyay et al. (1994)	9*7	4
P2	Sarker and Balan (1996)	8*7	4
P3	Mosier (1985)	10*10	5
P4	Seifoddini and Djassemi (1996)	11*7	4
P5	Jayakrishnan and Narendran (1998)	12*10	5
P6	Logendran (1990)	14*7	4
P7	Chandrasekharan and Rajagopalan (1986)	20*8	4
P8	Wen et al. (1996)	22*11	4

Table 3.1. The selected problems and the maximum number of machines per cell.

Part	Machine Number						
Type	1	2	3	4	5	6	7
1	1	1	0	0	1	0	1
2	1	0	0	1	1	0	0
3	0	0	1	1	0	1	0
4	1	1	0	0	0	0	1
5	1	0	0	0	0	0	1
6	0	0	0	1	1	0	0
7	0	0	1	0	0	1	0
8	0	0	1	1	0	1	0
9	1	0	1	0	1	0	1

Table 3.2. The original machine-part matrix from Mukhopadhyay et al. (1994).

Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

Part	1	2	3	4	5	6	7	8	9
D	299	291	239	210	203	281	248	260	273
S	14	14	11	10	10	14	12	13	13

Table 3.3. Generated demand and selling price for Mukhopadhyay et al. (1994).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between $10 and $15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =1.6593		CPU Time =25 (sec)
Cell	Parts	Machines
1	1,2,4,5,9	1,2,5,7
2	3,6,7,8	3,4,6

Part Type	Machine Number 1	2	5	7	3	4	6
1	1	1	1	1			
2	1		1			1	
4	1	1		1			
5	1			1			
9	1		1	1	1		
3					1	1	1
6			1			1	
7					1		1
8					1	1	1

Table 3.4. MM solution for Mukhopadhyay et al. (1994).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part Type	Machine Number 1	2	3	4	5	6	7
1	0	1	0	1	0	0	0
2	1	0	0	0	0	0	1
3	1	0	0	0	1	0	1
4	0	0	1	0	0	1	0
5	1	0	0	0	1	0	1
6	0	1	0	1	0	0	0
7	0	0	1	0	0	0	0
8	0	0	0	0	1	0	1

Table 3.5. The original machine-part matrix from Sarker and Balan (1996).

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

Part	1	2	3	4	5	6	7	8
D	299	291	239	210	203	281	248	260
S	14	14	11	10	10	14	12	13

Table 3.6. Generated demand and selling price for Sarker and Balan (1996).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between $10 and $15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =2.9733		CPU Time =21 (sec)
Cell	Parts	Machines
1	1,2,3,5,6,8	1,2,5,7
2	4,7	3,4,6

Part Type	Machine Number 1	2	5	7	3	4	6
1		1				1	
2	1			1			
3	1		1	1			
5	1		1	1			
6		1				1	
8			1	1			
4					1		1
7					1		

Table 3.7. MM solution for Sarker and Balan (1996).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part Type	Machine Number 1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	1	0	0
3	0	0	0	0	1	1	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	0	1	0
6	1	0	0	0	0	0	1	0	1	0
7	0	0	1	0	0	0	1	1	1	0
8	0	0	0	0	0	1	0	0	0	1
9	0	1	1	1	0	0	0	1	0	0
10	0	1	1	1	0	0	0	0	0	0

Table 3.8. The original machine-part matrix from Mosier (1985).

Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

Part	1	2	3	4	5	6	7	8	9	10
D	299	291	239	210	203	281	248	260	273	255
S	14	14	11	10	10	14	12	13	13	11

Table 3.9. Generated demand and selling price for Mosier problem (1985).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between \$10 and \$15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =1.9912		CPU Time =180.00 (sec)
Cell	Parts	Machines
1	1,4,5,6,7	1,7,9
2	3,8	5,6,10
3	2,9,10	2,3,4,8

Part Type	Machine Number 1	7	9	5	6	10	2	3	4	8
1	1		1							
4	1									
5		1	1							
6	1	1	1							
7		1	1					1		1
3				1	1					
8					1	1				
2								1	1	1
9							1	1	1	1
10							1	1	1	

Table 3.10. MM solution for Mosier (1985).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part Type	Machine Number 1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	1	1	0	0	0	0
3	1	0	0	0	1	1	0
4	0	0	0	1	0	0	1
5	0	0	0	1	0	0	1
6	0	1	1	0	0	0	0
7	1	0	0	0	1	1	0
8	0	0	0	1	0	0	1
9	0	1	1	0	0	0	0
10	0	0	0	1	0	0	1
11	1	0	0	0	1	1	0

Table 3.11. The original machine-part matrix from Seifoddini and Djassemi (1996).

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

Part	1	2	3	4	5	6	7	8	9	10	11
D_i	299	291	239	210	203	281	248	260	273	225	291
S_i	14	14	11	10	10	14	12	13	13	11	14

Table 3.12. Generated demand and selling price for Seifoddini and Djassemi (1996).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between $10 and $15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =2.7584		CPU Time =26 (Sec)
Cell	Parts	Machines
1	3,7,11	1,5,6
2	1,2,6,9	2,3
3	4,5,8,10	4,7

Part Type	Machine Number 1	5	6	2	3	4	7
3	1	1	1				
7	1	1	1				
11	1	1	1				
1				1	1		
2				1	1		
6				1	1		
9				1	1		
4						1	1
5						1	1
8						1	1
10						1	1

Table 3.13. MM solution for Seifoddini and Djassemi (1996).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part Type	Machine Number 1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	1	0	0	0	0	1
2	1	0	0	1	1	0	0	0	0	1
3	1	0	0	1	1	1	1	0	0	1
4	0	0	0	0	0	1	1	0	1	0
5	0	1	1	0	1	1	0	1	0	1
6	1	0	1	0	0	1	1	1	0	0
7	0	0	0	0	1	0	0	0	0	1
8	0	1	0	0	0	1	0	0	0	1
9	1	1	0	0	0	1	1	1	1	0
10	0	0	0	0	0	0	0	1	0	0
11	1	1	0	1	0	0	1	0	0	1
12	1	1	1	0	1	1	0	1	0	1

Table 3.14. The original machine-part matrix from Jaykrishnan and Narendran (1998).

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

Part	1	2	3	4	5	6	7	8	9	10
D	299	291	239	210	203	281	248	260	273	299
S	14	14	11	10	10	14	12	13	13	14
Part	11	12								
D	291	239								
S	14	11								

Table 3.15. Generated demand and selling price for Jaykrishnan and Narendran (1998).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between $10 and $15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =0.8070		CPU Time =74 (sec)
Cell	Parts	Machines
1	1,2,3,4,5,6,7,8,9,10,11,12	1,5,6,8,10
2		2,3,4
3		9
		7

Part Type	Machine Number 1	5	6	8	10	2	3	4	9	7
1	1	1	0	0	1	0	0	1	0	0
2	1	1	0	0	1	0	0	1	0	0
3	1	1	1	0	1	0	0	1	0	1
4	0	0	1	0	1	0	0	0	0	1
5	0	1	1	1	1	1	1	0	0	0
6	1	0	1	1	0	0	1	0	0	1
7	0	1	0	0	1	0	0	0	0	0
8	0	0	1	0	1	1	0	0	0	0
9	1	0	1	1	0	1	0	0	1	1
10	0	0	0	1	0	0	0	0	0	0
11	1	0	0	0	1	1	0	1	0	1
12	1	1	1	1	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

Table 3.16. MM solution for Jaykrishnan and Narendran (1998).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part Type	Machine Number 1	2	3	4	5	6	7
1	0	1	0	0	0	1	1
2	1	0	0	0	1	0	0
3	0	0	0	1	0	0	0
4	0	1	0	0	0	0	1
5	1	1	0	0	0	0	1
6	0	1	0	0	0	1	1
7	0	1	0	1	0	0	0
8	1	0	0	0	1	1	0
9	1	0	0	1	0	0	1
10	0	0	1	0	0	0	1
11	0	0	1	1	0	0	0
12	0	0	1	1	0	0	0
13	0	0	1	0	0	0	0
14	1	0	0	0	1	0	0

Table 3.17. The original machine-part matrix from Logendran (1990).

Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

Part	1	2	3	4	5	6	7	8	9	10
D	299	291	239	210	203	281	248	260	273	255
S	14	14	11	10	10	14	12	13	13	11
Part	11	12	13	14						
D	290	291	239	218						
S	14	10	11	13						

Table 3.18. Generated demand and selling price for Logendran (1990).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between \$10 and \$15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =1.9156		CPU Time =101(sec)
Cell	**Parts**	**Machines**
1	1,4,5,6,10	2,6,7
2	3,7,11,12,13	3,4
3	8,9,14	1
4	2	5

Part	Machine Number						
Type	2	6	7	3	4	1	5
1	1	1	1				
4	1		1				
5	1		1			1	
6	1	1	1				
10			1	1			
3					1		
7	1				1		
11				1	1		
12				1	1		
13				1			
8		1				1	1
9			1		1	1	
14						1	1
2						1	1

Table 3.19. MM solution for Logendran (1990).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part Type	Machine Number 1	2	3	4	5	6	7	8
1	0	0	0	0	1	1	0	0
2	1	0	1	0	0	0	0	0
3	1	1	0	1	0	0	1	1
4	0	1	0	1	0	0	1	1
5	0	0	0	0	1	1	0	0
6	0	1	0	1	1	0	1	1
7	0	1	0	1	0	0	1	1
8	1	0	1	0	0	0	0	0
9	1	0	1	0	0	1	0	0
10	0	0	0	1	1	1	0	0
11	1	0	1	0	0	0	1	0
12	0	0	0	0	1	1	1	0
13	1	0	1	0	0	0	0	0
14	1	1	1	0	0	0	0	0
15	0	0	0	0	1	1	0	0
16	1	0	1	0	0	0	0	0
17	1	0	1	0	1	0	0	0
18	1	0	0	1	0	0	1	1
19	1	0	1	0	0	0	0	0
20	0	1	0	1	0	1	1	1

Table 3.20. The original machine-part matrix from Chandrasekharan and Rajagopalan (1986).

Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

Part	1	2	3	4	5	6	7	8	9	10
D	299	291	239	210	203	281	248	260	273	255
S	14	14	11	10	10	14	12	13	13	11
Part	11	12	13	14	15	16	17	18	19	20
D	290	291	239	218	233	281	245	230	263	225
S	14	10	11	13	10	14	12	12	13	10

Table 3.21. Generated demand and selling price for Chandrasekharan and Rajagopalan problem (1986).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between \$10 and \$15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =1.2992		CPU Time =306(sec)
Cell	Parts	Machines
1	1,4,5,6,7,10,12,15,20	2,4,5,6
2	2,3,8,9,11,13,14,16,17,18,19	1,3,7,8

Part Type	Machine Number 2	4	5	6	1	3	7	8
1			1	1				
4	1	1						
5			1	1				
6	1	1	1				1	1
7	1	1						
10		1	1	1				
12			1	1			1	
15			1	1				
20	1	1		1			1	1
2					1	1		
3	1	1			1		1	1
8					1	1		
9					1	1		
11					1	1		
13					1	1		
14	1				1	1		
16					1	1		
17			1		1	1		
18	1	1					1	1
19					1	1		

Table 3.22. MM solution for Chandrasekharan and Rajagopalan (1986).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part	Machine Number										
Type	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	1	1	0	0	0	0	1	0
2	1	0	0	1	1	0	0	0	0	1	0
3	1	0	0	1	1	1	1	0	0	1	0
4	0	0	0	0	0	1	1	0	1	0	1
5	0	1	1	0	1	1	0	1	0	1	0
6	1	0	1	0	0	1	1	1	0	0	0
7	0	0	0	0	1	0	0	0	0	1	0
8	0	1	0	0	0	1	0	0	0	1	0
9	1	1	0	0	0	1	1	1	1	0	1
10	0	0	0	0	0	0	0	1	0	0	1
11	1	1	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	0	1	0
13	0	1	1	0	0	0	0	1	0	0	0
14	0	0	0	0	0	0	1	1	1	0	0
15	1	0	0	1	1	0	0	0	0	1	0
16	1	1	1	1	1	0	1	0	0	1	0
17	0	1	0	0	0	1	1	1	1	0	1
18	0	0	0	0	0	0	1	1	1	1	1
19	1	1	1	1	0	1	0	1	0	0	0
20	1	0	0	1	1	0	0	0	0	0	0
21	1	0	0	1	1	0	0	0	0	0	0
22	1	0	0	1	1	0	0	0	0	0	0

Table 3.23. The original machine-part matrix from Wen et al. (1996).

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

Part	1	2	3	4	5	6	7	8	9	10	11
D	299	291	239	210	203	281	248	260	273	299	291
S	14	14	11	10	10	14	12	13	13	14	14
Part	12	13	14	15	16	17	18	19	20	21	22
D	239	210	203	281	248	260	273	210	270	210	239
S	11	10	10	14	12	13	13	12	11	11	14

Table 3.24. Generated demand and selling price for Wen et al. (1996).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between \$10 and \$15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Productivity =0.7957		CPU Time =776 (sec)
Cell	**Parts**	**Machines**
1	1,2,3,7,11,15,16,20,21,22	1,4,5,10
2	4,5,6,8,9,12,13,14,17,18,19	2,6,7,8
3	10	11
4		3,9

Part Type	Machine Number 1	4	5	10	2	6	7	8	11	3	9
1	1	1	1	1	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0	0	0
3	1	1	1	1	0	1	1	0	0	0	0
7	1	1	0	1	1	0	1	0	0	0	0
11	1	1	0	1	0	0	1	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0	0
16	1	1	1	1	1	0	1	0	0	1	0
20	1	1	1	0	0	0	0	0	0	0	0
21	1	1	1	0	0	0	0	0	0	0	0
22	1	1	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	0	1	0	1
5	0	0	1	1	1	1	0	1	0	1	0
6	1	0	0	0	0	1	1	1	0	1	0
8	0	0	0	1	1	1	0	0	0	0	0
9	1	0	0	0	1	1	1	1	1	0	1
12	1	0	1	1	1	1	0	1	0	0	0
13	0	0	0	0	1	0	0	1	0	1	0
14	0	0	0	0	0	0	1	1	0	0	1
17	0	0	0	0	1	1	1	1	1	0	1
18	0	0	0	1	0	0	1	1	1	0	1
19	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	1	1	0	0
										0	0

Table 3.25. MM solution for Wen et al. (1996).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

3.6 Sensitivity Analysis

The applicability of the developed model to find the optimum solution was tested using eight problems selected from the literature in section 3.5. A change in the input factors may result in a new solution. Since the effect of changing the values of the input parameters on the optimum solution is investigated in this section. Two problems of different size and structure are selected to study the effect of varying the parameter values on the final solution. The first one is a well-structured problem presented by Seifoddine and Djassemi (1996). The second problem is an ill structured problem presented by Logendran (1990).

The effect of the maximum number of machines allowable in each cell for both problems is studied by decreasing and increasing the allowable number of machines per cell. The model is run for the different cases and the changes in the final solution are recorded and summarized in Tables 3.26 and 3.27. In the Seifoddine and Djassemi (1996) problem (Table 3.26), the results show that the final solution changes by decreasing the maximum number of machines in each cell by 2 and 3. However, it remains constant as the maximum number of machines in each cell increases. In the Logendran (1990) problem (Table 3.27), changes in the maximum number of machines allowable in each cell change the part families and machine groups as well as the objective function values except when the maximum number of machines allowable in each cell is decreased or increased by 1.

The effect in intra-cell movement cost is investigated by decreasing and increasing the original intra-cell cost per movement by 10%, 20%, 30%, 40%, 50% and 70%. The other parameters remain constant. The model is run and the cell configurations are determined and summarized in Tables 3.28 and 3.29. In both problems, decreasing or increasing the intra-cell movement cost does not change the cell configuration since the intra-cell cost is less than the inter-cell cost. However, it changes the objective function values.

The effect of varying the inter-cell movement cost on the final solution for the Logendran (1990) problem is studied by decreasing and increasing the inter-cell cost by 10%, 20%, 30%, 40%, 50% and 70%. The various cases are analyzed and the results are presented in Table 3.30. The results show that changing the inter-cell cost does not affect the solution as long as a smaller value is assigned to the intra-cell movement cost. The effect of changes in inter-cell cost for the Seifoddine and Djassemi (1996) problem is not studied since there is no inter-cell movement.

The effect of production volume is investigated by decreasing and increasing the original number of units of a randomly selected part by 10%, 20%, 30%, 40%, 50% and 70%. Part 1 is randomly selected for the Seifoddine and Djassemi (1996) problem while part 9 is also randomly selected for the Logendran (1990) using a FORTRAN subroutine adapted from Microsoft FORTRAN and presented in Appendix 1. The changes in the optimum solution are recorded and presented in Tables 3.31 and 3.32. The results obtained show that there is no change in the final solution by decreasing or increasing the production rate in the Seifoddine and Djassemi (1996) problem. In the Logendran (1990) problem, the results show that decreasing or increasing the production rate by 50% or more changes the final solution.

M_{max}	Cell	Part families	Machine groups	Objective function	Cell configuration
6	1	3, 7, 11	1, 5, 6	2.7584	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
5	1	3, 7, 11	1, 5, 6	2.7584	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
Original 4	1	3, 7, 11	1, 5, 6	2.7584	Original
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
3	1	3, 7, 11	1, 5, 6	2.7584	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
2	1	1, 2, 6, 5	2, 3	2.1570	Change
	2	3, 7, 11	5, 6		
	3	4, 5, 8, 10	4, 7		
	4		1		
1	1	3, 7	5	1.2048	Change
	2		1		
	3	4, 8, 10	4		
	4	1, 2, 9	2		
	5	9	3		
	6	6	7		
	7		6		

Table 3.26. Effect of changing the maximum number of machines for Seifoddine and Djassemi (1996).

M_{max}	Cell	Part families	Machine groups	Objective function	Cell configuration
7	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1, 2, 3, 4, 5, 6, 7	2.4595	Change
6	1	1, 2, 3, 4, 5, 6, 8, 9, 14	1, 2, 5, 6, 7	2.4595	Change
	2	3, 7, 10, 11, 12, 13	3, 4		
5	1	1, 4, 5, 6	2, 6, 7	1.9156	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 10, 11, 12, 13	3, 4		
Original 4	1	1, 4, 5, 6	2, 6, 7	1.9156	Original
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 10, 11, 12, 13	3, 4		
3	1	1, 4, 5, 6	2, 6, 7	1.9156	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 10, 11, 12, 13	3, 4		
2	1	1, 4, 5, 6, 10	2, 7	1.7166	Change
	2	2, 8, 14	1, 5		
	3	3, 7, 9, 11, 12, 13	3, 4		
	4	1, 6, 7	6		
1	1		2	1.2092	Change
	2		1		
	3	4, 5, 9, 10	7		
	4	3	4		
	5	11, 12, 13	3		
	6	2, 8, 14	5		
	7		6		

Table 3.27. Effect of changing the maximum number of machines per cell for Logendran (1990).

M_{max}	Cell	Part families	Machine groups	Objective function	Cell configuration
-70%	1	3, 7, 11	1, 5, 6	9.1947	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-50%	1	3, 7, 11	1, 5, 6	5.5168	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-40%	1	3, 7, 11	1, 5, 6	4.5973	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-30%	1	3, 7, 11	1, 5, 6	3.9405	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-20%	1	3, 7, 11	1, 5, 6	3.4480	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-10%	1	3, 7, 11	1, 5, 6	3.0649	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
Original 2	1	3, 7, 11	1, 5, 6	2.7584	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
10%	1	3, 7, 11	1, 5, 6	2.5076	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
20%	1	3, 7, 11	1, 5, 6	2.2986	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
30%	1	3, 7, 11	1, 5, 6	2.12186	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
40%	1	3, 7, 11	1, 5, 6	1.9703	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
50%	1	3, 7, 11	1, 5, 6	1.8389	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
70%	1	3, 7, 11	1, 5, 6	1.4910	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		

Table 3.28. Effect of changing intra-cell handling cost for Seifoddine and Djassemi (1996).

Intra-cell Cost	Cell	Part families	Machine groups	Objective function	Cell configuration
-70%	1	1, 4, 5, 6, 10	2, 6, 7	3.7450	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-50%	1	1, 4, 5, 6, 10	2, 6, 7	2.9422	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-40%	1	1, 4, 5, 6, 10	2, 6, 7	2.6570	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-30%	1	1, 4, 5, 6, 10	2, 6, 7	2.4228	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-20%	1	1, 4, 5, 6, 10	2, 6, 7	2.2263	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-10%	1	1, 4, 5, 6, 10	2, 6, 7	2.0593	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
Original 2	1	1, 4, 5, 6, 10	2, 6, 7	1.9156	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
10%	1	1, 4, 5, 6, 10	2, 6, 7	1.6153	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
20%	1	1, 4, 5, 6, 10	2, 6, 7	1.681	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
30%	1	1, 4, 5, 6, 10	2, 6, 7	1.5841	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
40%	1	1, 4, 5, 6, 10	2, 6, 7	1.4975	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
50%	1	1, 4, 5, 6, 10	2, 6, 7	1.4201	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
70%	1	1, 4, 5, 6, 10	2, 6, 7	1.2869	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		

Table 3.29. Effect of changing intra-cell handling cost for Logendran (1990).

Inter-cell Cost	Cell	Part families	Machine groups	Objective function	Cell configuration
-70%	1	1, 4, 5, 6, 10	2, 6, 7	2.4294	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-50%	1	1, 4, 5, 6, 10	2, 6, 7	2.2565	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-40%	1	1, 4, 5, 6, 10	2, 6, 7	2.1789	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-30%	1	1, 4, 5, 6, 10	2, 6, 7	2.1065	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-20%	1	1, 4, 5, 6, 10	2, 6, 7	2.0388	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-10%	1	1, 4, 5, 6, 10	2, 6, 7	1.9753	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
Original 4.6	1	1, 4, 5, 6, 10	2, 6, 7	1.9156	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
10%	1	1, 4, 5, 6, 10	2, 6, 7	1.8606	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
20%	1	1, 4, 5, 6, 10	2, 6, 7	1.8064	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
30%	1	1, 4, 5, 6, 10	2, 6, 7	1.7564	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
40%	1	1, 4, 5, 6, 10	2, 6, 7	1.7090	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
50%	1	1, 4, 5, 6, 10	2, 6, 7	1.6642	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
70%	1	1, 4, 5, 6, 10	2, 6, 7	1.5811	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		

Table 3.30. Effect of changing inter-cell handling cost for Logendran (1990).

Demand	Cell	Part families	Machine groups	Objective function	Cell configuration
-70%	1	3, 7, 11	1, 5, 6	2.7067	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-50%	1	3, 7, 11	1, 5, 6	2.7223	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-40%	1	3, 7, 11	1, 5, 6	2.7296	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-30%	1	3, 7, 11	1, 5, 6	2.7370	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-20%	1	3, 7, 11	1, 5, 6	2.7442	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
-10%	1	3, 7, 11	1, 5, 6	2.7514	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
Original	1	3, 7, 11	1, 5, 6	2.7584	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
10%	1	3, 7, 11	1, 5, 6	2.7652	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
20%	1	3, 7, 11	1, 5, 6	2.7720	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
30%	1	3, 7, 11	1, 5, 6	2.7786	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
40%	1	3, 7, 11	1, 5, 6	2.7849	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
50%	1	3, 7, 11	1, 5, 6	2.7917	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		
70%	1	3, 7, 11	1, 5, 6	2.8037	No change
	2	1, 2, 6, 9	2, 3		
	3	4, 5, 8, 10	4, 7		

Table 3.31. Effect of changing production rate of part no.1 for Seifoddine and Djassemi (1996).

Demand	Cell	Part families	Machine groups	Objective function	Cell configuration
-70%	1	1, 4, 5, 6, 7, 10	2, 6, 7	2.0695	Change
	2	2, 8, 9, 14	1, 5		
	3	3, 11, 12, 13	3, 4		
-50%	1	1, 4, 5, 6,7, 10	2, 6, 7	2.0210	Change
	2	2, 8, 9, 14	1, 5		
	3	3, 11, 12, 13	3, 4		
-40%	1	1, 4, 5, 6, 10	2, 6, 7	1.9978	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-30%	1	1, 4, 5, 6, 10	2, 6, 7	1.9763	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-20%	1	1, 4, 5, 6, 10	2, 6, 7	1.9555	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
-10%	1	1, 4, 5, 6, 10	2, 6, 7	1.9348	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
Original	1	1, 4, 5, 6, 10	2, 6, 7	1.9156	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
10%	1	1, 4, 5, 6, 10	2, 6, 7	1.8970	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
20%	1	1, 4, 5, 6, 10	2, 6, 7	1.8784	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
30%	1	1, 4, 5, 6, 10	2, 6, 7	1.8611	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
40%	1	1, 4, 5, 6, 10	2, 6, 7	1.8444	No change
	2	2, 8, 9, 14	1, 5		
	3	3, 7, 11, 12, 13	3, 4		
50%	1	1, 4, 5, 6, 7, 10	2, 6, 7	1.8282	Change
	2	2, 8, 9, 14	1, 5		
	3	3, 11, 12, 13	3, 4		
70%	1	1, 4, 5, 6, 7, 10	2, 6, 7	1.7968	Change
	2	2, 8, 9, 14	1, 5		
	3	3,11, 12, 13	3, 4		

Table 3.32. Effect of changing production rate of part no.9 for Logendran (1990).

CHAPTER 4
SIMULATED ANNEALING APPROACH

Chapter 3 presents a solution to the cell formation problem in CM systems, a 0-1 integer programming model aims at maximizing the ratio of output to the total material handling cost is developed. The results show the ability of the model to solve different problems. In addition, the model offers some advantages over some of the existing models including consideration of production volumes and selling prices, and its ability to form part families and machine cells simultaneously. Furthermore, the developed model has the ability to determine the optimum number of manufacturing cells. However, the model needs visual inspection to identify part families and machine groups, and requires long computational times to solve large-scale problems. This chapter presents a simulated annealing algorithm to overcome the limitations associated with the integer-programming model developed in chapter 3.

4.1 Introduction

An optimization problem can be described by identifying a set of solutions $X=(x_1, x_2, x_3,\ldots,x_n)$ that optimizes some function $f(X)$ subject to certain constraints. An optimum solution is one that gives the best possible objective function (Johnson *et al.*, 1989). There may be more than one optimum solution; for instance, in Figure 4.1 there is more than one optimum solution on the search space. These solutions are known as local optimum solutions.

Local search algorithms start with an initial solution *S*. Some changes in the initial solution are made to obtain a new solution S'. The new solution S' is then compared with the initial solution S. If S' is better than S, the new solution is accepted; otherwise the new solution is rejected. Figure 4.2 illustrates the local search process. These algorithms are simple to implement and quick to execute; however, they terminate at the first local optimum because only movements that improve the objective function are accepted (Mukhopadhay *et al.*, 1998; Oliveria and Ferreira, 1993).

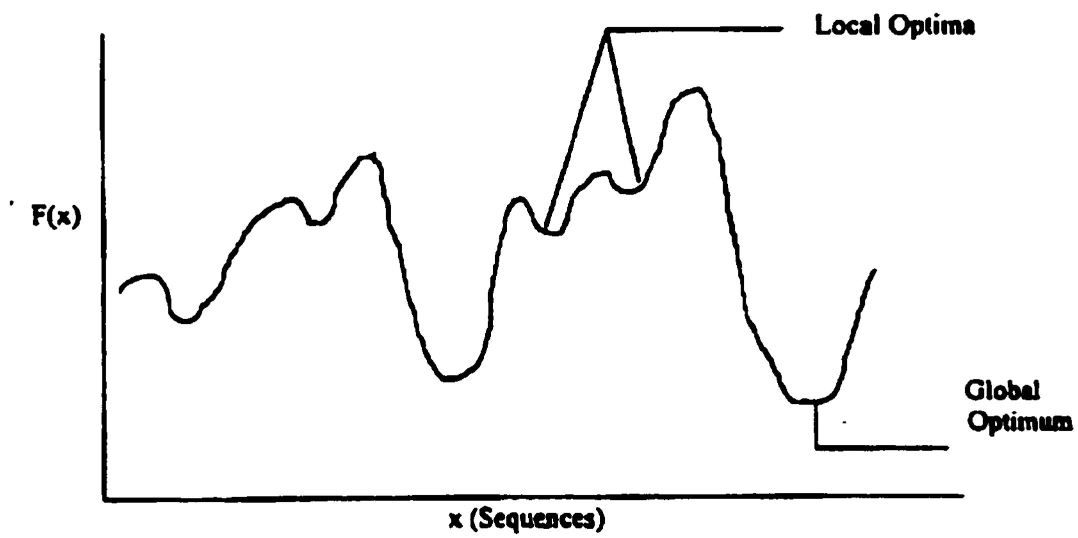


Figure 4.1. Local and global optimum points (Mukhopadhyay *et al.*, 1998).

Note: x= Variable; F(x)= Value of the objective function

1. Get an initial solution S.
2. While there is an untested neighbor of S do the following:
   2.1. Let S' be an untested neighbor of S.
   2.2. If cost (S') < cost (S), set S=S'.
3. Return to 2.

Figure 4.2. Local search procedure (Johnson *et al.*, 1989).

On the other hand, simulated annealing algorithms accept the worst solution to avoid getting in a local optimum with a certain probability. SA algorithms have been successfully applied to solve the cell formation problem in CM systems (Chen & Srivastava 1994; Adil *et al.*, 1996). The solution obtained by SA does not depend on the initial part families and machine groups and has an objective function closer to the global solution (Adil *et al.*, 1996).

4.2 Concepts and Basic Structure

Simulated annealing algorithm as an optimization approach was introduced by Metroplis *et al.* (1953) and described by several researchers (Kirkpatric *et al.*, 1983; Laarhoven and Arts; 1987; Adil *et al.*, 1996; Chen and Srivastava, 1994). It is based on the analogy between finding an optimal solution in solving optimization problems and finding a low energy state in the annealing process of solids. Annealing is a physical process for obtaining a low energy state of a solid in two steps:

1. The solid is heated to the melting point.
2. The temperature of the solid is decreased slowly, allowing it to come to thermal equilibrium at each temperature.

In physical systems, temperature has a physical meaning while in SA temperature is a control parameter. The analogy between the annealing process and an optimization problem is summarized in Figure 4.3. Simulating the annealing process is very similar to an optimization problem. Metropolis *et al.* (1953) simulated the annealing process and introduced an algorithm known as the Metropolis algorithm (Figure 4.4).

The idea for the Metropolis algorithm is that given the current state of the solid with energy E_1, a change mechanism is applied to generate the next state with energy E_2 by a small displacement of a randomly selected particle. If the energy is decreased, i.e. $\Delta E=(E_2- E_1)<0$, the new configuration has low energy and is accepted as a starting point for the next move. If $\Delta E>0$; the new state may be accepted. In physical systems, jumps to a higher energy state usually occur; however, these moves are controlled by the temperature T at that state (Rutenbar, 1989). At higher temperatures the probability of large uphill moves is high; it is small at lower temperatures. Metropolis *et al.* (1953) modeled these moves with a Boltzmann distribution in which the probability of an uphill move of size ΔE at temperature T is given by $e^{(-\Delta E/T)}$. A random number R in (0,1) is then generated and the move is only accepted if $R \le e^{(-\Delta E/T)}$.

Annealing process	Optimization problem
Temperature	A control parameter that controls the number of iterations.
The state of a physical system	The solution of the problem
The energy of the state	The objective function value
The ground state (minimum energy)	Optimal solution

Figure 4.3. Analogy between annealing process and optimization problem.

Simulated annealing algorithms start by choosing an initial feasible solution (current solution) and calculating the objective function. A new solution (neighbor solution) is then randomly generated and the objective function is calculated. The change in the objective function is then calculated and evaluated. These procedures continue until the stopping criteria are met. Sridhar and Rajendran (1993) presented a simple simulated annealing algorithm as shown in Figure 4.5.

```
M= number of moves to attempt
T= current temperature
For m=1 to M
    Generate a random move, e.g., move a particle;
    Evaluate the change in energy, ΔE;
    If (ΔE<0) (i.e. downhill move: accept it)
    Else
    (uphill move)
    accept with probability P=EXP(-ΔE/T)
    update configuration if accepted
end
```

Figure 4.4. Metropolis algorithm.

```
1. Get an initial solution S.
2. Get an initial temperature T>0
3. While not yet frozen do the following:
   3.1. Perform the following loop L times
        3.1.1. Pick a random neighbor S' of S
        3.1.2. Let Δ=cost(S') -cost (S)
        3.1.3. If Δ<=0 then set S=S'.
        3.1.4. If Δ>0 then set S=S' with probability exp (-Δ/T).
4. Set T=αT.
Return to 3.
```

Figure 4.5. SA procedure (Sridhar and Rajendran, 1993).

4.3 The Algorithm

In this section, a simulated annealing algorithm is developed to solve the cell formation problem in CM systems. A summary of the proposed algorithm is presented in Figure 4.6. The proposed algorithm is coded in FORTRAN 77 using a Pentium 120 MHz computer. The FORTRAN program for this algorithm is presented in Appendix 3. The following section explains how the proposed algorithm works.

1. Input

   1.1. Read the input data: number of parts P, number of machines M, number of cells C, demand D, selling price S, inter-cell handling cost EMC, intra-cell handling cost IMC, and maximum number of machines allowed in each cell M_{max}.

   1.2. Define the annealing parameters: initial temperature T_0, final temperature T_f, decrement factor α, and the number of iterations at each temperature level NIRAT, where NIRAT= K*NM, where K is a constant, called the size factor that controls the cooling rate, and NM is the number of machines in the problem.

2. Initialization

   2.1. Set the counter i =0

   2.2. Find an initial solution. Calculate the total material handling cost by assuming that each part is moving through all cells to complete the required operations. Then calculate the initial objective function OBJ_0 of the current manufacturing cells.

   2.3. Let the objective function OBJ_0=OBJ=BOBJ, where OBJ is the current objective function and BOBJ is the best objective function found so far.

3. Set the number of iterations at each temperature level IR=0

4. Generate a neighboring solution

   4.1. Set IR=IR+1

4.2. Randomly select a machine to move to a randomly selected cell and re-calculate the objective function of the current solution. Set the new solution = NOBJ

5. Evaluation

5.1. If NOBJ $\geq$ OBJ, then set OBJ=NOBJ, and go to step 6

5.2. If NOBJ $\geq$ BOBJ, then set BOBJ=NOBJ, and go to step 6

5.3. If NOBJ< OBJ, then let the probability of acceptance of a new solution equal $e^{(-\Delta z/T)}$, where Δz = NOBJ-OBJ. Choose a random number R in (0,1); if R $\leq e^{(-\Delta z/T)}$, then set OBJ=NOBJ. Otherwise reject the solution and go to step 6.

6. If the number of iterations IR<NIRAT, then go to 4.1. Otherwise go to 7.

7. Adjust the temperature, set the current temperature T = α T and go to step 8

8. If T $\leq T_f$, then go to 9. Otherwise go to step 3.

9. Part assignment

   Calculate the number of machines required by each part in all cells and assign the part to the cell with the maximum number of machines and go to 10.

10. Print the best solution obtained and stop.

4.4 Illustrative Examples

The developed algorithm is tested using the same eight problems presented in section 3.5. Based on the application of simulated annealing, the performance of the simulated annealing algorithm depends on the annealing parameters (Boctor, 1996; Adil *et al.*, 1996). These parameters include initial temperature, number of neighboring solutions generated at each temperature level, decrement factor, and conditions to stop the searching process. Therefore, the first step in implementing the developed algorithm is the determination of the annealing parameter values.

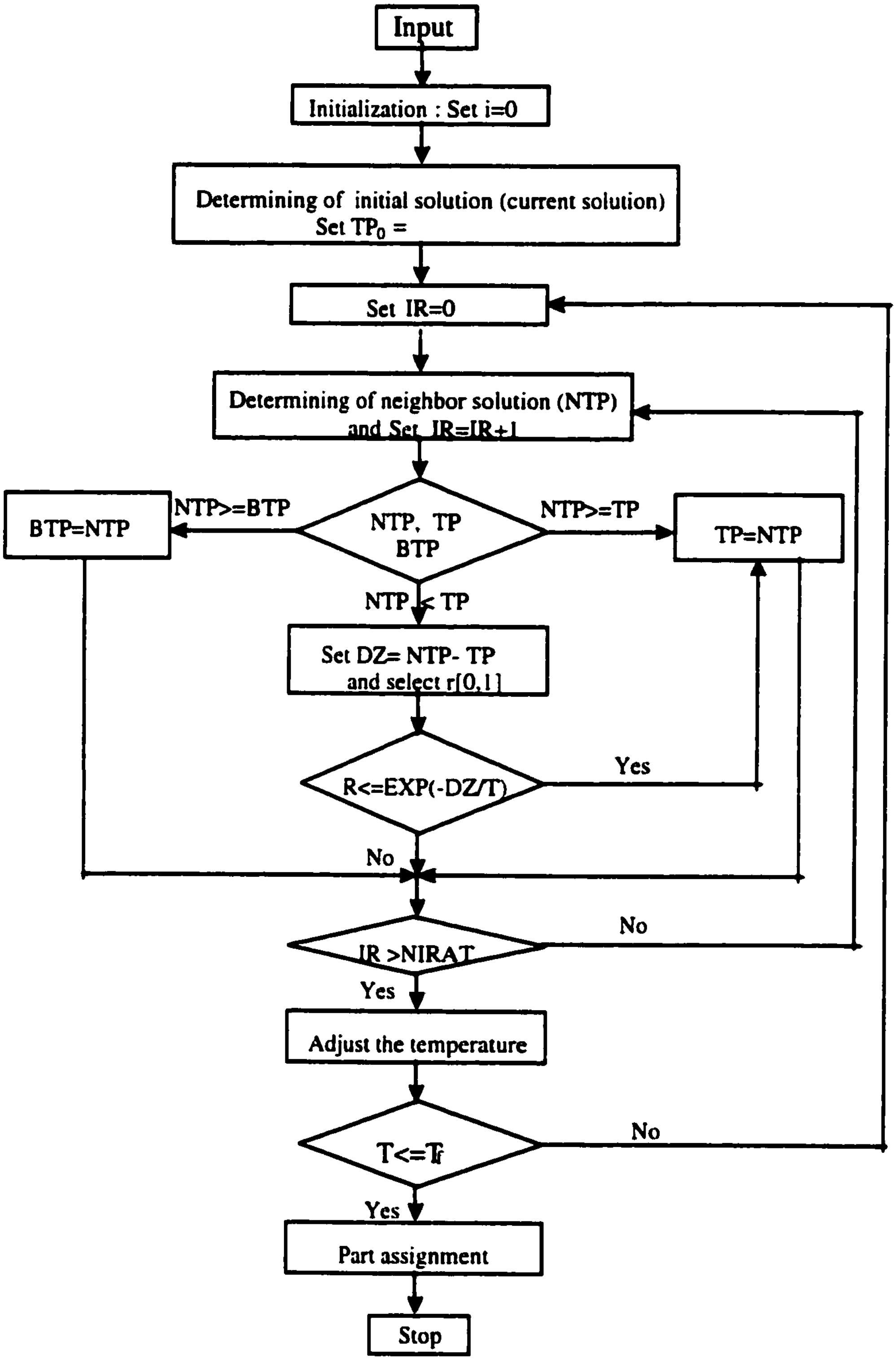


Figure 4.6. Flow chart of the SA algorithm.

4.4.1 Selection of Initial Temperature

In the physical analogy, the initial temperature T_0 of a solid should be raised to the melting point to ensure that all particles are randomly arranged in a liquid state. This means that at the beginning of the annealing process, the search process can reach all possible configurations. Similarly in SA, the initial temperature T_0 should be raised to a sufficient level to allow the algorithm to accept a high percentage of new solutions (Chen and Srivastava, 1994; Adil *et al.*, 1996). Starting at high initial temperatures allows the algorithm to get out of the local optimum. However, a high initial temperature might increase the computational time. Hence, the initial temperature T_0 should be selected so that the probability of acceptance of a new solution will be sufficient. Kirkpatrick *et al.* (1983) stated that the value of T_0 should be selected to ensure that at least 80% of new solutions are accepted. That can be described mathematically in the form $e^{(-\Delta z/T)} \geq 0.8$; where Δz is the difference in the objective function, and T is the initial temperature.

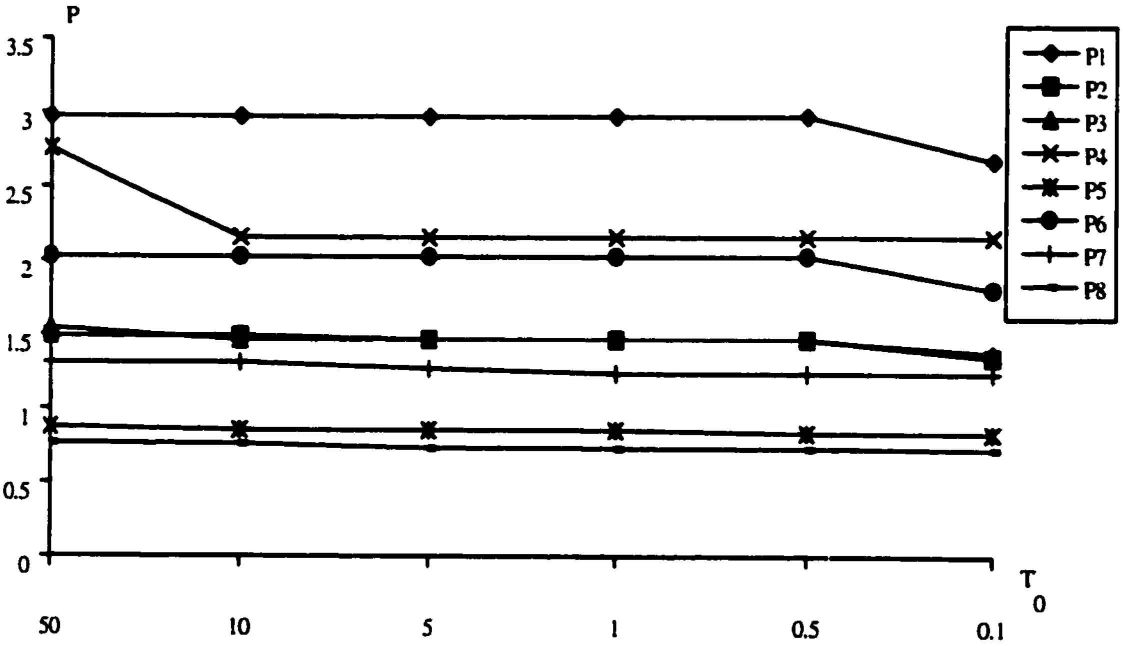
To determine the most suitable value for the starting temperature, different values for T_0 are assumed (Table 4.1). For this experiment the other annealing parameters are assumed as follows: $\alpha = 0.99$, $T_f = 0.05$, and K=1. The initial probability of acceptance of a new solution is calculated after the first temperature. A stopping temperature is assumed to terminate the algorithm after the first temperature. The stopping condition is then removed and the algorithm is run to find the CPU time and the objective function.

The results are presented in Table 4.1 and plotted against initial temperature in Figure 4.7. The results are selected based on both CPU time and the value of the objective function. It is clear that for problems 1 and 6 the initial temperatures $T_0 = 50$, 10, 5, 1, and 0.5 give the same objective function values. However, $T_0 = 0.5$ has the shortest CPU time and therefore is selected as the best T_0 for both problems 1 and 6. Clearly, for problems 2 and 7, the most promising initial temperatures are $T_0 = 50$ and 10. The initial temperature $T_0 = 50$ is eliminated because of its long computational time, and therefore $T_0 = 10$ is selected for both problems. Similarly, the best T_0 value for problems 3, 4, 5, and 8 is $T_0 = 50$. These values are indicated in bold in Table 4.1. Figure 4.7 shows the effect of T_0 on both productivity and running time. As shown in the figure, T_0 has different values for different problems, and therefore no standard initial temperature T_0 can be recommended.

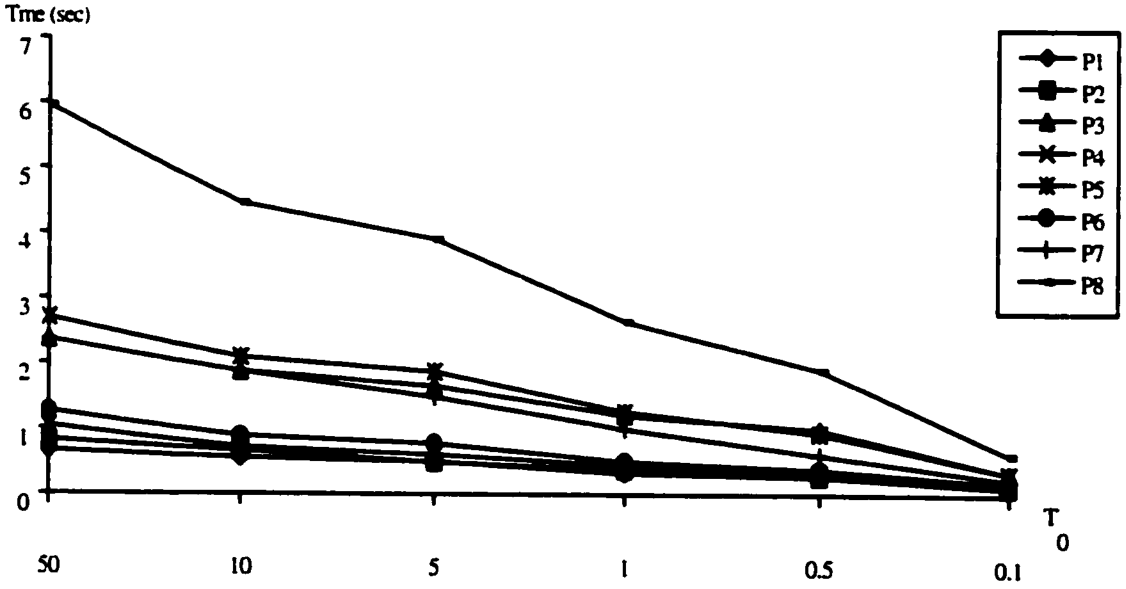
T_0		50	10	5	1	0.5	0.1
P1	IPA	0.99	0.95	0.91	0.65	0.39	0.30
	CPU	0.66	0.55	0.50	0.33	0.27	0.11
	OBJ	2.9733	2.9733	2.9733	2.9733	2.9733	2.6705
P2	IPA	0.99	0.99	0.98	0.91	0.83	0.26
	CPU	0.82	0.66	0.49	0.38	0.27	0.11
	OBJ	1.4880	1.4880	1.4627	1.4627	1.4627	1.3443
P3	IPA	0.99	0.99	0.98	0.93	0.88	0.54
	CPU	2.36	1.87	1.65	1.21	0.99	0.33
	OBJ	1.5414	1.4631	1.4631	1.4631	1.4631	1.3754
P4	IPA	0.99	0.97	0.96	0.81	0.66	0.42
	CPU	1.04	0.71	0.60	0.44	0.33	0.11
	OBJ	2.7584	2.1572	2.1572	2.1572	2.1572	2.1572
P5	IPA	0.99	0.99	0.98	0.93	0.88	0.56
	CPU	2.69	2.09	1.87	1.26	0.93	0.33
	OBJ	0.8708	0.8511	0.8511	0.8511	0.8321	0.8289
P6	IPA	0.99	0.98	0.96	0.85	0.73	0.12
	CPU	1.26	0.88	0.77	0.50	0.38	0.17
	OBJ	2.0265	2.0265	2.0265	2.0265	2.0265	1.8052
P7	IPA	0.99	0.99	0.99	0.95	0.91	0.67
	CPU	2.36	1.87	1.48	1.00	0.60	0.22
	OBJ	1.3096	1.3096	1.2668	1.2343	1.2343	1.2343
P8	IPA	0.99	0.99	0.98	0.94	0.91	0.81
	CPU	5.95	4.45	3.90	2.64	1.90	0.61
	OBJ	0.7634	0.7586	0.7283	0.7271	0.7271	0.7230

Table 4.1. Selection of initial temperature (T_0) for all problems.

Note: IPA= initial probabability of acceptance; CPU = computional time in seconds; OBJ = objective function



a) Productivity (P)



b) Computational time

Figure 4.7. The effect of initial temperature T_0 on both productivity and CPU time.

4.4.2 Selection of Cooling Rate

The cooling rate is controlled by both the decreasing factor (α) and the number of new solutions generated at each temperature level. The decreasing factor represents the rate at which the temperature is decreased. This parameter should be decreased slowly enough to allow the algorithm to search a wide area and to accept a relatively large number of new solutions. The number of new solutions at each temperature level is determined by multiplying the number of machines (NM) in the system by a constant called the size factor (k) (Chen and Srivastava, 1994). As the number of iterations increases, more new solutions are generated. This will increase the possibility of finding the global solution. However, it increases the running time, and therefore it is important to find the optimum values for both decreasing and size factors.

The effect of the size factor K on cooling rate is studied first. Different values for the size-factor are assumed with the decremental factor α set to 0.99. The values for the size factor (K) are assumed such that each value doubles the previous one. The initial temperature value for each problem obtained in the previous experiment is used. For the purpose of this experiment the final temperature T_f is set to 0.05. The algorithm is run and the total productivity and computational time are calculated. The evaluation of the results obtained is based on the computational time and the value of the objective function.

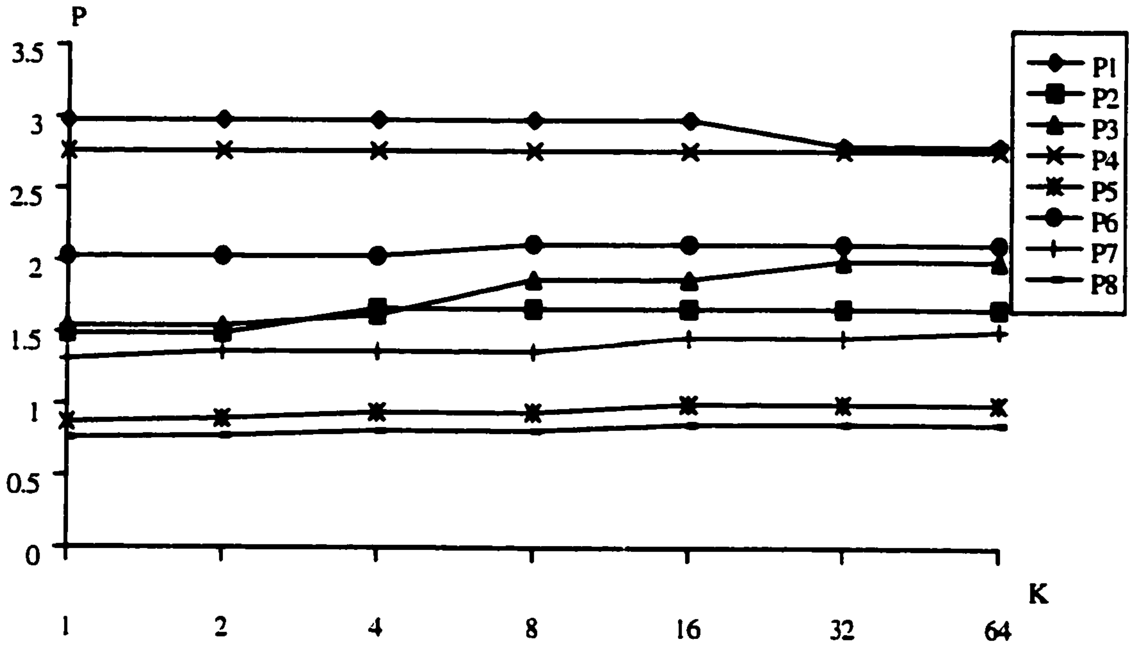
Table 4.2 and Figure 4.8 show that the size factor K=1 results in the highest total productivity and shortest CPU time for both problems 1 and 4 and is therefore selected as the best value. For problem 2, the size factors K=4, 8, 16, 32, and 64 are the most promising ones. However, K= 4 has the shortest computational time and is selected as the best value. The most promising K for problems 3 and 8 is K=32 with the highest total productivity and shortest running time. Similarly, for problems 5, 6, and 7 the best K values giving the highest total productivity and shortest running time are 16, 8, and 16 respectively. It can be seen from Figure 4.8 that the value of size factor K varies from one problem to another based on the size and complexity of each problem. Hence, no standard value can be recommended.

The effect of the decreasing factor α on cooling rate is studied by assuming different values for α such that each value squares the pervious value ($0.99 \leq \alpha \leq 0.85$). The final temperature T_f is set at 0.05. The values of other parameters obtained in the previous experiments are used in this experiment. Table 4.3 presents the results obtained and Figure 4.9 shows the effect of the decremental factor α on both total productivity and running time. The results clearly illustrate that the decremental factor $\alpha = 0.98$ is the best value for problem 1. For all other problems $\alpha = 0.99$ provides the best values in both total productivity and computational time and is therefore selected. The best values for α are indicated in bold in Table 4.3.

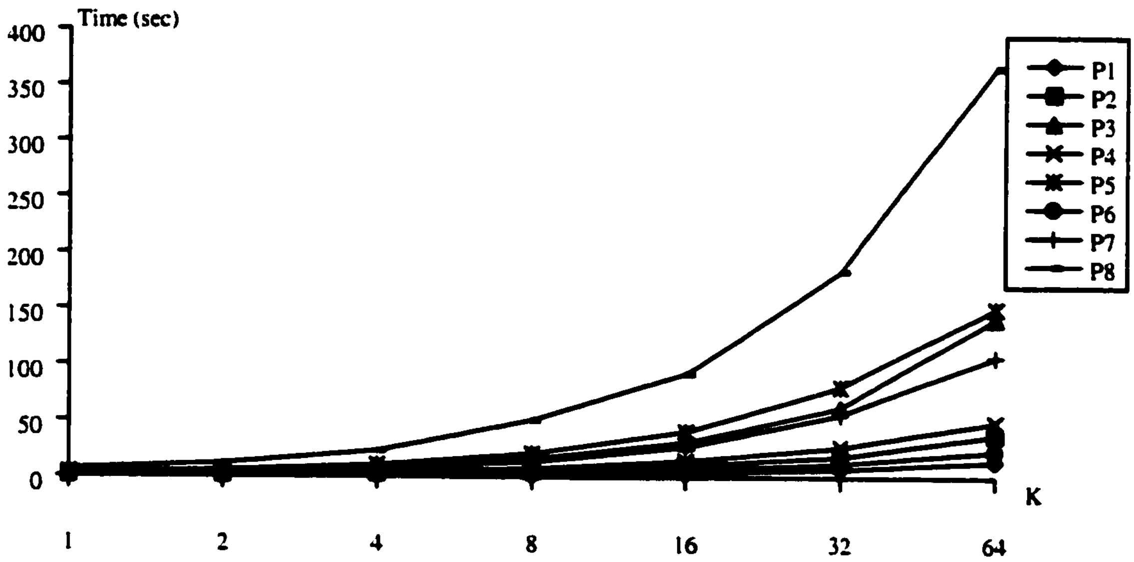
K		1	2	4	8	16	32	64
P1	CPU	0.27*	0.66	0.93	1.50	3.10	6.99	14.28
	OBJ	2.9733	2.9733	2.9733	2.9733	2.9733	2.7933	2.7933
P2	CPU	0.82	1.54	2.69	5.05	10.36	18.20	37.79
	OBJ	1.4880	1.4880	1.6595	1.6595	1.6595	1.6595	1.6595
P3	CPU	2.36	4.23	8.13	15.76	31.09	61.79	141.05
	OBJ	1.5414	1.5414	1.6170	1.8638	1.8638	1.9927	1.9927
P4	CPU	1.04	2.14	3.90	7.36	14.40	26.50	49.10
	OBJ	2.7584	2.7584	2.7584	2.7584	2.7584	2.7584	2.7584
P5	CPU	2.69	5.11	10.16	20.50	40.32	80.50	150.10
	OBJ	0.8708	0.8954	0.9389	0.9389	0.9970	0.9970	0.9970
P6	CPU	0.38	0.71	1.54	3.02	6.10	11.90	23.34
	OBJ	2.0265	2.0265	2.0265	2.1079	2.1079	2.1079	2.1079
P7	CPU	1.87	3.51	6.98	13.10	26.97	55.80	107.16
	OBJ	1.3096	1.3628	1.3628	1.3628	1.4569	1.4569	1.5067
P8	CPU	5.95	11.48	22.96	50.20	91.67	183.4	365.69
	OBJ	0.7634	0.7742	0.8111	0.8111	0.8566	0.8608	0.8608

Table 4.2. Selection of the size factor (K) for all problems.

Note: CPU = computional time in seconds; OBJ = objective function



a) Productivity (P)



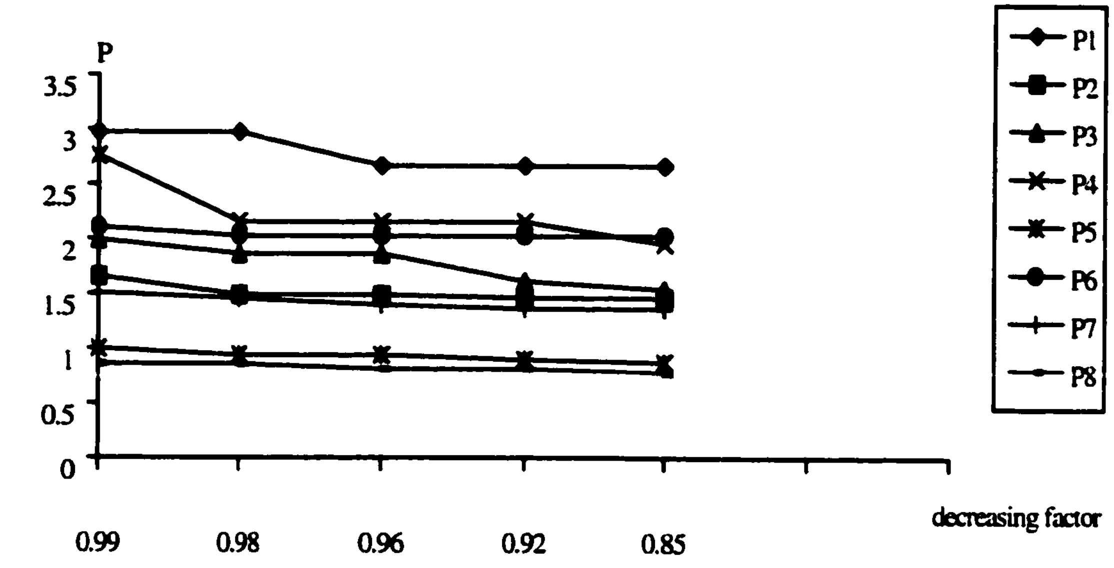
b) Computational time

Figure 4.8. The effect of the size factor K on both productivity and CPU time.

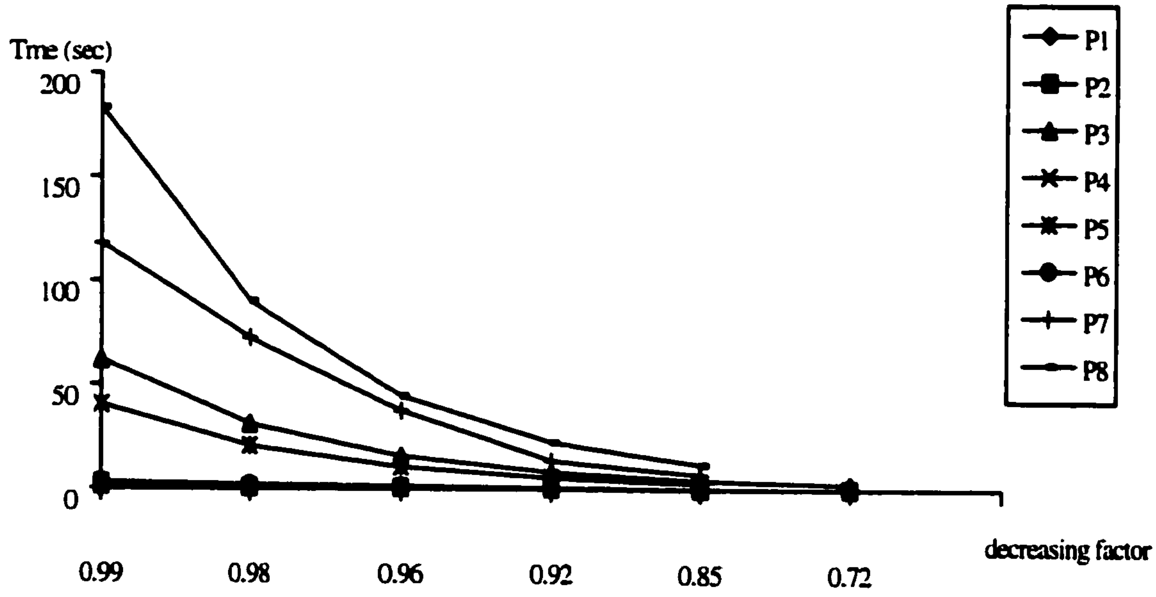
α		0.99	0.98	0.96	0.92	0.85
P1	CPU	0.27*	0.17	0.11	0.08	0.06
	OBJ	2.9733	2.9733	2.6705	2.6705	2.6705
P2	CPU	2.69	1.59	0.77	0.44	0.22
	OBJ	1.6595	1.4880	1.4880	1.4627	1.4627
P3	CPU	61.79	31.06	15.65	7.91	4.33
	OBJ	1.9927	1.8638	1.8638	1.6170	1.5414
P4	CPU	1.04	0.65	0.27	0.17	0.11
	OBJ	2.7584	2.1572	2.1572	2.1572	1.9566
P5	CPU	40.32	20.48	10.44	5.39	3.02
	OBJ	0.9970	0.9383	0.9383	0.8954	0.8708
P6	CPU	3.02	1.93	0.99	0.55	0.24
	OBJ	2.1079	2.0265	2.0265	2.0265	2.0265
P7	CPU	107.76	72.20	40.96	13.52	7.08
	OBJ	1.5064	1.4569	1.3990	1.3628	1.3628
P9	CPU	183.40	90.10	44.51	22.47	11.92
	OBJ	0.8609	0.8566	0.8111	0.8111	0.7742

Table 4.3. Selection of the decreasing factor (α) for all problems.

Note: CPU = computional time in seconds; OBJ = objective function



a) Productivity (P)



b) Computational time

Figure 4.9. The effect of the decreasing factor α on both productivity and CPU time.

4.4.3 Selection of Stopping Conditions

The stopping conditions refer to the parameters used to terminate the searching process. Various stopping criteria have been suggested in the literature: (1) when the objective function value satisfies a lower bound (Laarhoven & Arts, 1987); (2) if the objective function value does not change for a certain number of iterations (Chen & Srivastava, 1994); (3) if the number of iterations exceeds the maximum allowed number of iterations (Adil *et al.*, 1996). However, these stopping criteria may not allow the annealing process to continue until the system is frozen.

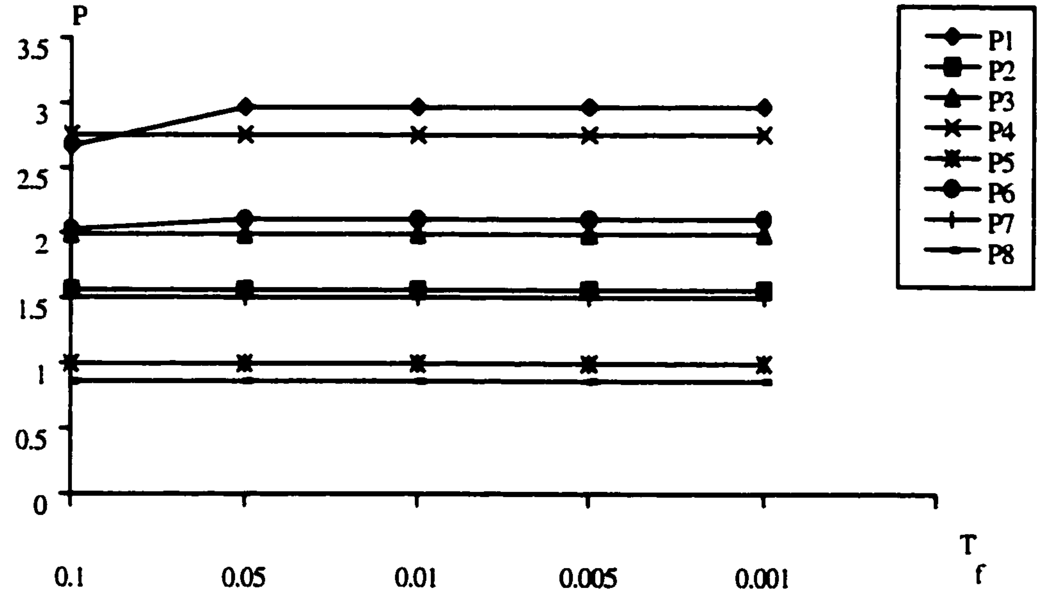
In this research, the final temperature T_f is used to terminate the algorithm to ensure the frozen state of the liquid and ensure enough search space. The algorithm is stopped if the temperature level becomes less than or equal to the final temperature T_f.

Table 4.4 presents the results obtained and Figure 4.10 shows those results plotted against final temperature. By comparing the results on the bases of total productivity and running time, T_f =0.05, 0.01, and 0.005 are the most promising final temperatures for problems 1, 6, and 8. However, T_f = 0.05 is selected since it results in the highest productivity and shortest running time. Similarly, the final temperature T_f = 0.1 is selected as the best stopping temperature for problems 2, 3, 4, and 5. As was the case for the previous parameters, no standard stopping temperature value is recommended.

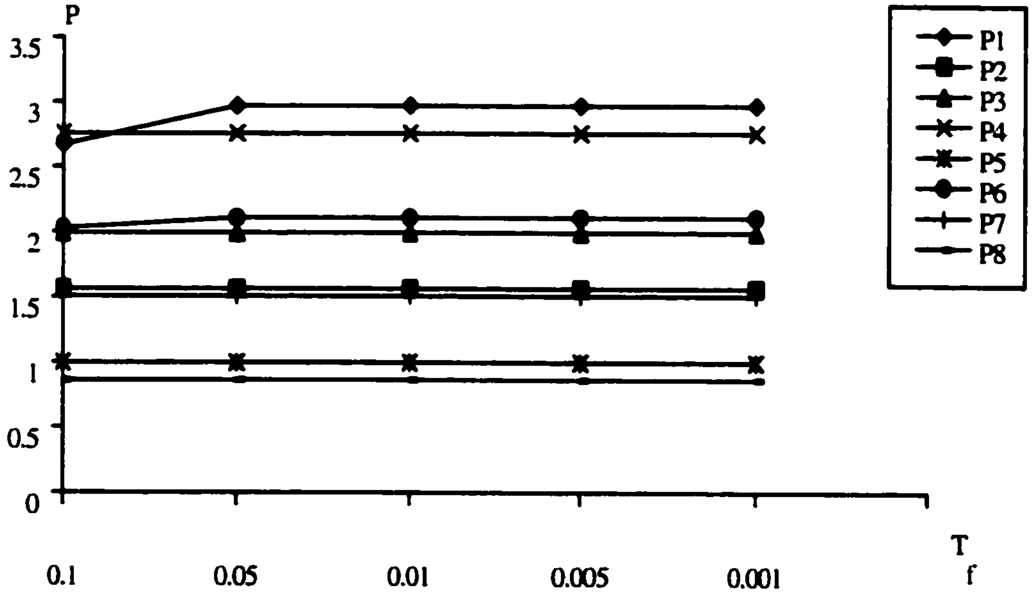
T_f		0.1	0.05	0.01	0.005	0.001
P1	CPU	0.11*	0.17	0.22	0.33	0.71
	OBJ	2.6705	2.9733	2.9733	2.9733	2.9733
P2	CPU	2.09	2.69	3.08	3.41	4.12
	OBJ	1.5616	1.5616	1.5616	1.5616	1.5616
P3	CPU	55.42	61.79	76.12	82.38	110.10
	OBJ	1.9927	1.9927	1.9927	1.9927	1.9927
P4	CPU	0.82	1.04	1.10	1.64	2.25
	OBJ	2.7584	2.7584	2.7584	2.7584	2.7584
P5	CPU	36.36	40.32	52.40	54.26	63.33
	OBJ	0.9970	0.9970	0.9970	0.9970	0.9970
P6	CPU	2.03	3.02	5.11	6.31	8.02
	OBJ	2.0265	2.1079	2.1079	2.1079	2.1079
P7	CPU	94.36	107.76	188.12	232.26	332.19
	OBJ	1.5067	1.5067	1.5067	1.5067	1.5067
P8	CPU	135.60	183.40	226.68	245.19	298.21
	OBJ	0.8602	0.8608	0.8608	0.8608	0.8608

Table 4.4. Selection of stopping conditions (T_f) for all problems.

Note: CPU = computational time in seconds; OBJ = objective function

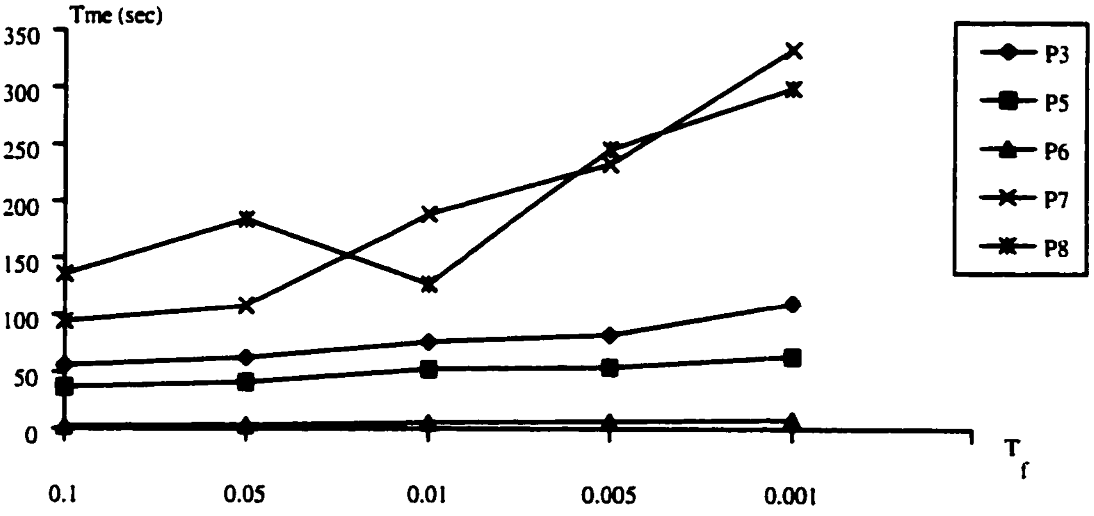


a) Productivity (P)



b) Computational time

Figure 4.10. The effect of the final temperature T_f on both productivity and CPU time.



b) Computational time

Figure 4.10 (cont.). The effect of the final temperature T_f on productivity and CPU time.

4.4.4 Results

Based on the experiments conducted in the previous sections, the values of the simulated annealing algorithm parameters presented in Table 4.5 were determined to be the best for each problem. Clearly, no strong correlation is evident in the results and therefore no standard values are recommended. The results obtained using these annealing parameters for the eight problems are presented in Tables 4.5 to 4.13.

Problem Number	Parameter			
	T_0	K	α	T_f
P1	0.5	1	0.98	0.05
P2	10	4	0.99	0.1
P3	50	32	0.99	0.1
P4	50	1	0.99	0.1
P5	50	16	0.99	0.1
P6	0.5	8	0.99	0.05
P7	10	64	0.99	0.1
P8	50	32	0.99	0.1

Table 4.5. Best values for the annealing parameters.

Note: T_0= initial temperature; K=size-factor; α=decremental factor; T_f=stopping temperature

Productivity =2.9733	CPU Time =0.16 (sec)	
Cell	Parts	Machines
1	1,4,6,7	2,3,4,6
2	2,3,5,8	1,5,7

Part Type	Machine Number 2	3	4	6	1	5	7
1	1		1				
4		1		1			
6	1		1				
7		1					
2					1		1
3					1	1	1
5					1	1	1
8						1	1

Table 4.6. SA solution for Sarker and Balan (1996)

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Productivity =1.6593		CPU Time =2.47 (sec)
Cell	Parts	Machines
1	1,4,5,9	1,2,7
2	2,3,6,7,8	3,4,5,6

Part Type	Machine Number 1	2	7	3	4	5	6
1	1	1	1				
4	1	1	1				
5	1		1				
9	1		1	1		1	
2	1				1	1	
3				1	1		1
6					1	1	
7				1			1
8				1	1		1

Table 4.7. SA solution for Mukhopadhyay *et al.* (1994)

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Productivity =1.9927		CPU Time =55.42 (sec)
Cell	Parts	Machines
1	2,7,9,10	2,3,4,8
2	8	10
3	3	5,6
4	1,4,5,6	1,7,9

Part Type	Machine Number 2	3	4	8	10	5	6	1	7	9
2		1	1	1						
7		1		1					1	1
9	1	1	1	1						
10	1	1	1							
8					1			1		
3						1	1			
1								1		1
4								1		
5									1	1
6								1	1	1

Table 4.8. SA solution for Mosier (1985).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Productivity =2.7584		CPU Time =1.26 (sec)
Cell	Parts	Machines
1	3,7,11	1,5,6
2	1,2,6,9	2,3
3	4,5,8,10	4,7

Part Type	Machine Number 1	5	6	2	3	4	7
3	1	1	1				
7	1	1	1				
11	1	1	1				
1				1	1		
2				1	1		
6				1	1		
9				1	1		
4						1	1
5						1	1
8						1	1
10						1	1

Table 4.9. SA solution for Seifoddini and Djassemi (1996).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Productivity =0.99701		CPU Time =36.36 (sec)
Cell	Parts	Machines
1	3,4,5,7,8,9,11	2,6,7,10
2		9
3	1,2,6,10,12	1,3,4,5,8

Part Type	Machine Number 2	6	7	10	9	1	3	4	5	8
3		1	1	1		1		1	1	
4		1	1		1					
5	1	1		1			1		1	1
7				1					1	
8	1	1		1						
9	1	1	1		1	1				1
11	1		1	1		1		1		
1				1		1		1	1	
2				1		1		1	1	
6		1	1			1	1			1
10										1
12		1		1		1	1		1	1

Table 4.10. SA solution for Jaykrishhnan and Narendran (1998).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Productivity =2.1079		CPU Time = 3.40 (sec)
Cell	Parts	Machines
1	2,8,14	1,5,6
2	1,3,4,5,6,7,9,10,11,12,13	2,3,4,7

Part Type	Machine Number 1	5	6	2	3	4	7
2	1	1					1
8	1	1	1				
14	1	1					
1			1	1			1
3						1	
4				1			1
5	1			1			1
6			1	1			
7				1		1	
9	1					1	1
10					1		1
11					1	1	
12					1	1	
13					1		

Table 4.11. SA solution for Logendran (1990).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Productivity =1.5067		CPU Time = 94.36 (sec)
Cell	Parts	Machines
1	1,4,5,6,7,10,12,15,18,20	4,5,6,8
2	2,3,8,9,11,13,14,16,17,19	1,2,3,7

Part Type	Machine Number 4	5	6	8	1	2	3	7
1		1	1					
4	1			1		1		1
5		1	1					
6	1	1		1		1		1
7	1			1		1		1
10	1	1	1					
12		1	1					1
15		1	1					
18	1				1			1
20	1		1	1		1		1
2					1		1	
3	1			1	1	1		1
8					1		1	
9			1		1		1	
11					1		1	1
13					1		1	
14					1	1	1	
16					1		1	
17		1			1		1	
19					1		1	

Table 4.12. SA solution for Chandrasekharan and Rajagopalan (1986).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Productivity =0.8608		CPU Time =183.40 (sec)
Cell	Parts	Machines
1	1,2,3,5,7,11,15,16,20,21,22	4,5,10
2	4,8,9,14,17,18	2,6,7,9
3	6,10,12,13,19	1,3,8,11

Part Type	Machine Number 4	5	10	2	6	7	9	1	3	8	11
1	1	1	1					1			
2	1	1	1					1			
3	1	1	1		1	1		1			
5		1	1	1	1					1	
7		1	1								
11	1		1	1		1		1			
15	1	1	1					1			
16	1	1	1	1		1		1	1		
20	1	1						1			
21	1	1						1			
22	1	1						1			
4					1	1	1				1
8			1	1	1						
9				1	1	1	1	1		1	1
14						1	1			1	
17				1	1	1	1			1	1
18			1			1	1			1	1
6					1	1		1	1	1	
10										1	1
12		1	1		1			1	1	1	
13				1					1	1	
19	1			1	1			1	1	1	

Table 4.13. SA solution for Wen *et al.* (1996).

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

4.5 Comparison between MM and SA Algorithms

A comparison of the mathematical model (MM) with the simulated annealing (SA) algorithm is presented in this section. The results of applying both models to the eight selected problems are summarized in Table 4.14 and illustrated graphically in Figures 4.11 and 4.12. For the small size problems 1, 2, 3, and 4, both models provide the same productivity value. However, the SA took less computational time to find the solution. The following paragraphs present a brief analysis for problems 5, 6, 7, and 8.

Problem 5: It is clear that the SA outperforms the MM in both productivity and computational time. The productivity value determined using the MM is 0.807, whereas it is 0.997 as determined by the SA algorithm, a 23.5% improvement. The MM produces 4 manufacturing cells, and the SA produces 3 cells with different machine groups. An increase in the number of cells increases the number of inter-cell movements, which in turn decreases the system productivity.

Problem 6: There is a clear difference between both models in solving this problem. The SA outperforms the MM in both productivity and computational time. The productivity value obtained by the SA is 10% higher than that obtained by the MM. The MM identifies 3 manufacturing cells while the SA identifies only 2 cells. Increasing the number of cells results in increased inter-cell cost and decreased productivity value. The SA model showed better results in productivity value and computational time.

Problem 7: The results show that the SA performed better than the MM in both productivity and computational time. Both models formed 2 manufacturing cells; however, the decomposition of part families and machine groups is different.

Problem 8: The SA outperforms the MM in both productivity and computational time. The productivity value determined using the MM is 0.7957; the value using the SA is 0.8608, an 8% increase. The MM formed 4 manufacturing cells, while the SA formed 3 cells.

Prob No.	Source	CPU (sec)		Productivity	
		MM	SA	MM	SA
P1	Sarker and Balan (1996)	21	0.16	2.9733	2.9733
P2	Mukhopadhyay *et al.* (1994)	25	2.47	1.6593	1.6593
P3	Mosier (1985)	180	55.42	1.9927	1.9927
P4	Seifoddini and Djassemi (1996)	26	1.26	2.7584	2.7584
P5	Jayakrishnan and Narendran (1998)	74	36.3	0.8070	0.9970
P6	Logendran (1990)	101	3.40	1.9156	2.1079
P7	Chandrasekharan and Rajagopalan (1986)	306	94.36	1.2992	1.5067
P8	Wen *et al.* (1996)	776	183.40	0.7957	0.8608

Table 4.14. A comparison of SA with MM.

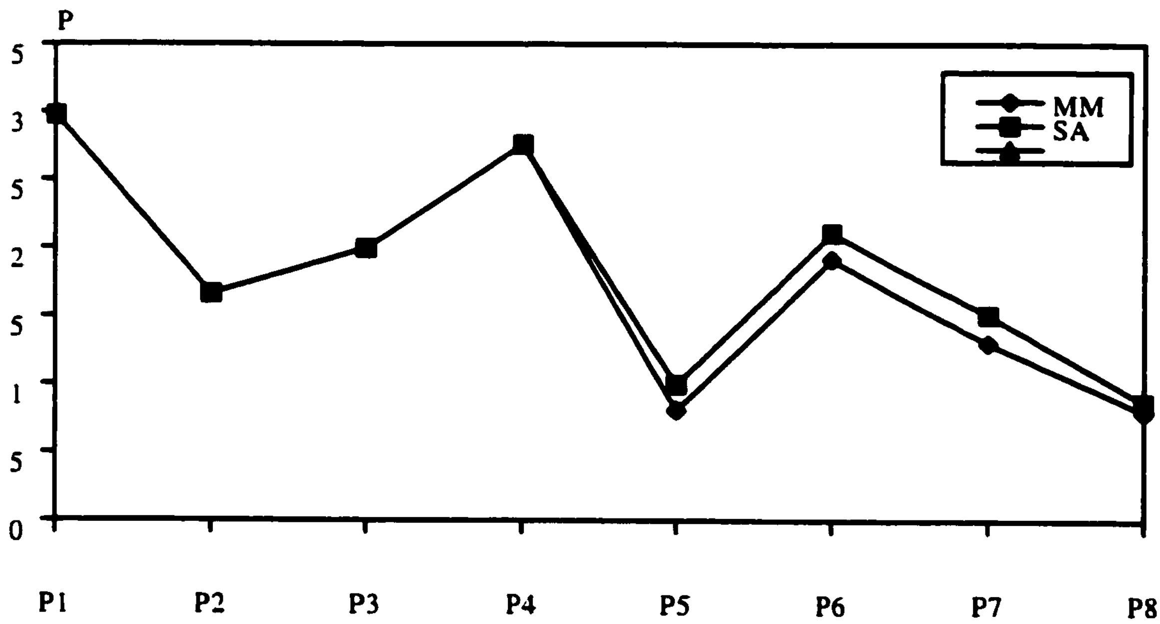


Figure 4.11. Productivity values obtained by both models.

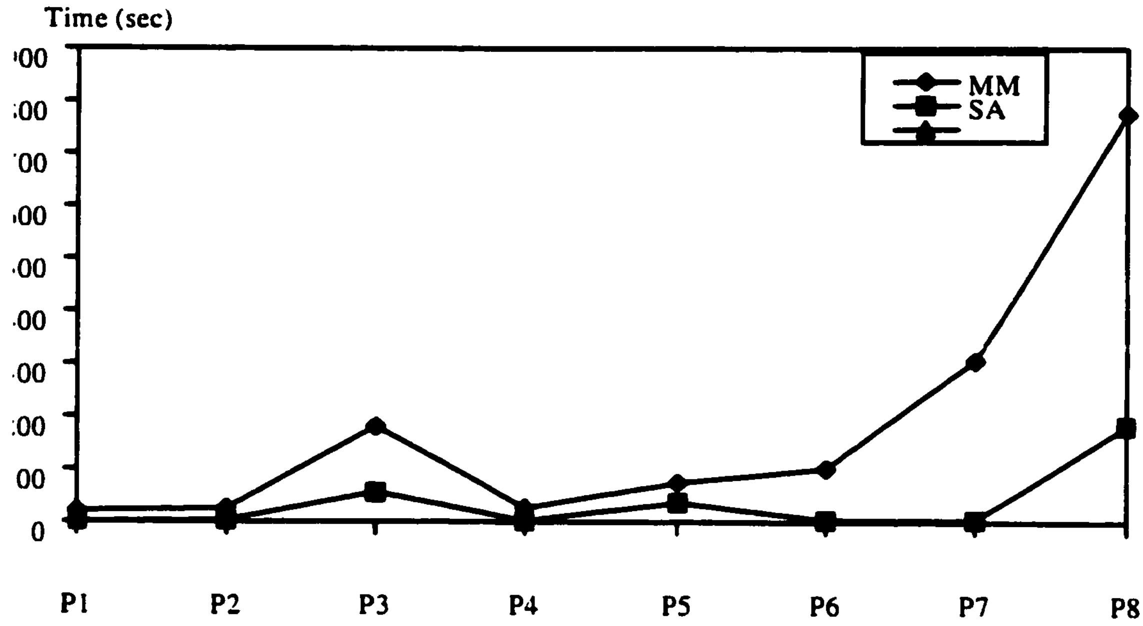


Figure 4.12. Computational time taken by both models.

4.6 Performance of the SA Algorithm for Large problems

The ability of the developed SA algorithm to solve large-scale problems is discussed in this section. Two problems of different size and structure are selected. The intra and inter-cell costs are assumed to be \$2 and \$4.6 per movement respectively in both problems. Information regarding annual demand and selling price is generated between 200 and 300, and 10 and 15 respectively for both problems. This is done using a FORTRAN subroutine adapted from Microsoft FORTRAN and presented in Appendix 1.

The first problem (P9) is a 30 part by 20 machine problem obtained by duplicating the 15 part by 10 machine problem considered by Chan and Milner (1982). The initial part-machine matrix is presented in Table 4.15. Information regarding annual demand

and selling price is presented in Table 4.16. It is assumed that the maximum number of machines per cell is 7. The second problem (P10) is a problem of 43 parts and 16 machines considered by King (1982). Table 4.17 presents the initial part-machine matrix. Information regarding annual demand and selling price is presented in Table 4.18. The maximum number of machines per cell is assumed to be 6.

The experiments conducted in section 4.3 are used to determine the suitable values for the annealing parameters for both problems P9 and P10. Table 4.19 presents the CPU time and objective function at different initial temperatures. Clearly, for P9 the initial temperatures T_0=5, 1 and 0.5 give the same objective function but the initial temperature 0.5 has the shortest CPU time and therefore is selected as the best T_0 for P9. Similarly, the best T_0 for P10 is $T_0 = 1$. These values are indicated in bold in Table 4.19. The effect of the size factor K is studied and the results are presented in Table 4.20. The size factor K=16 results in the highest productivity value and shortest CPU time for both problems and is therefore selected as the best value. Table 4.21 presents the effect of the cooling rate α on both productivity and CPU time. It is clear that α = 0.99 and 0.98 are the best cooling rates for P9 and P10 respectively. The effect of the stopping temperature T_f on both productivity and running time is presented in Table 4.22. Clearly, the most promising T_f for P9 is $T_f = 0.1$ and for P10 is $T_f = 0.05$. The previous results are summarized in Table 4.23. The following paragraphs present a brief analysis for each of these problems.

Problem 9: The developed SA shows its ability to solve this problem. Table 4.24 presents the final part families and machine groups as well as the optimum number of cells. The optimum solution was obtained in 436.60 seconds with an objective function value of 0.4700. The model produces 4 part families and 8 machine groups.

Problem 10: The SA finds the optimum solution for this problem in 389.86 seconds with an objective function value of 1.1326 (Table 4.25). The solution contains 6 part families and 6 machine groups.

Part	Machine number																			
type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0
2	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0
3	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
4	0	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0
5	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	1	1	0
7	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	0
8	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
9	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
10	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
11	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	0	0
12	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0
13	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
14	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
15	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
16	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0
17	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0
18	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
19	0	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0
20	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
21	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	1	1	0
22	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	0
23	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
24	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
25	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
26	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	0	0
27	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0

Table 4.15. The original machine-part matrix for the first problem (P9). Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

28	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
29	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
30	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1

Table 4.15 (Cont.). The original machine-part matrix for the first problem (P9).

Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

Part	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D	299	291	239	210	203	281	248	260	273	255	291	243	241	219	248
S	14	14	11	10	10	14	12	13	13	11	14	12	12	10	12
Part	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
D	212	200	227	203	270	257	210	234	209	203	283	214	209	206	258
S	10	10	11	10	13	12	10	11	10	10	14	10	10	10	12

Table 4.16. Generated demand and selling price for the first problem (P9).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between \$10 and \$15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Part	Machine Number															
Type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0
2	1	0	0	0	0	1	0	1	1	0	1	0	0	1	0	1
3	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
5	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
7	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
9	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	0
10	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
12	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
13	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0
14	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
17	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
18	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
19	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
21	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0
22	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
23	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0

Table 4.17. The original machine-part matrix for the second problem (P10).

24	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0
25	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
27	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0
28	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0
29	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
31	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
32	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1
33	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
34	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
35	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
36	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
37	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1
38	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1
39	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
40	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
41	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
42	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1
43	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0

Table 4.17 (Cont). The original machine-part matrix for the second problem (P10).

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

Part	1	2	3	4	5	6	7	8	9	10	11	12	13	14
D	299	291	239	210	203	281	248	260	273	255	291	243	241	219
S	14	14	11	10	10	14	12	13	13	11	14	12	12	10
Part	15	16	17	18	19	20	21	22	23	24	25	26	27	28
D	248	212	200	227	203	270	257	210	234	209	203	283	214	209
S	12	10	10	11	10	13	12	10	11	10	10	14	10	10
Part	29	30	31	32	33	34	35	36	37	38	39	40	41	42
D	206	258	269	248	260	221	231	245	240	290	239	256	240	270
S	10	12	13	12	13	11	11	12	12	14	11	12	12	10
Part	43													
D	243													
S	14													

Table 4.18. Generated demand and selling price for the second problem (P10).

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between \$10 and \$15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

T_0		50	10	5	1	0.5	0.1
(P9)	IPA	0.97	0.93	0.91	0.55	0.21	0.11
	CPU	40.67	31.75	25.10	18.30	14.25	4.18
	OBJ	0.4351	0.4351	0.5259	0.5259	0.5259	0.4259
(P10)	IPA	0.99	0.99	0.99	0.98	0.96	0.93
	CPU	28.12	20.10	18.78	12.31	9.56	3.02
	OBJ	1.0385	1.0385	1.0385	1.0385	1.0102	0.9864

Table 4.19. Selection of initial temperature (T_0) for both problems.

K		1	2	4	8	16	32	64
P9	CPU	31.67	1.02.89	2.05.34	4.10.20	8.20.10	16.40.14	34.10.12
	OBJ	0.4351	0.4351	0.4366	0.4607	0.4700	0.4700	0.4700
P10	CPU	12.31	25.50	52.10	1.37.77	3.15.92	7.20.11	13.02.36
	OBJ	1.0385	1.0385	1.0430	1.0664	1.0795	1.1015	1.1326

Table 4.20. Selection of the size factor (K) experiments for both problems.

α		0.99	0.98	0.96	0.92	0.85
P9	CPU	8.20.10	4.09.41	2.03.58	1.10.50	39.43
	OBJ	0.4700	0.4607	0.4366	0.4351	0.4351
P10	CPU	13.02.36	6.29.86	3.14.00	1.34.52	56.50
	OBJ	1.1326	1.1326	1.1015	1.0604	1.0430

Table 4.21. Selection of the decreasing factor (α) for both problems.

T_f		0.1	0.05	0.01	0.005	0.001
P9	CPU	7.16.60	8.20.10	10.50.20	12.00.29	14.50.25
	OBJ	0.4700	0.4700	0.4700	0.4700	0.4700
P10	CPU	5.10.50	6.29.86	9.20.16	11.52.10	14.35.75
	OBJ	1.1015	1.1326	1.1326	1.1326	1.1326

Table 4.22. Selection of stopping conditions (T_f) for both problems.

Problem Number	Parameter			
	T_0	K	α	T_f
P9	10	16	0.99	0.1
P10	1	64	0.98	0.05

Table 4.23. Best values for the annealing parameters.

Productivity =0.4700		CPU Time =436.60 (sec)
Cell	Parts	Machines
1		14,17
2		18
3	8,9,10,13,15,23,24,25,28,30	5,10,15
4	2,3,4,6,7,11,12,14,17,18,19,21,22, 26,27,29	1,4,6,7,9,16
5	5,20	2,12
6		13,19
7	1,16	3,8,11
8		20

Table 4.24. SA solution for the first problem (P9).

Productivity =1.1326		CPU Time =389.86 (sec)
Cell	Parts	Machines
1	6,35	14
2	1,12,13,25,26,39	10
3	4,7,8,10,17,18,28,32,34,36,37,38 40,42	7
4	2,3,9,11	2,3,6,9,13
5	15,19,20,21,22,23,24,27,29,30,31	1,4,8,11,12,16
	5,14,16,33,41,43	5,15

Table 4.25. SA solution for the second problem (P10).

CHAPTER 5

CELLULAR MANUFACTURING DESIGN CONSIDERING ALTERNATIVE PROCESS PLANS

This chapter presents a productivity model for solving the machine-part grouping problem in cellular manufacturing (CM) systems considering alternative process plans. The model aims at forming part families and machine groups simultaneously and selecting the process plan such that the system productivity is maximized. First, a 0-1 integer-programming model is developed to identify machine groups and part families simultaneously. Second, an efficient simulated annealing (SA) algorithm is developed to solve large-scale problems. This algorithm has several advantages over the existing ones. It forms part families and machine cells simultaneously, and considers production volume, selling price, maximum number of machines in each cell, and total material handling cost. Moreover, the proposed SA algorithm has the ability to determine the optimum number of manufacturing cells. Several problems of different size and complexity are used to test the performance of the developed models, and the results show the superiority of the SA algorithm over the integer-programming model in both productivity value and computational time.

5.1 Introduction

Cellular manufacturing is an application of group technology (GT) in which manufacturing systems are organized or re-arranged into manufacturing cells by grouping similar parts into part families and their required machines into manufacturing cells. The implementation of CM leads to increased sales and output, improved quality, decreased setup time, reduced work-in-process, reduced material handling cost as well as improved system productivity (Burbidge and Halsall, 1994). One problem in the design of CM systems is cell formation (CF). Solving the cell formation problem in CM systems leads to group similar parts into part families and their required machines into machine groups.

The CF problem has been studied and extensively reviewed in the literature (King and Nakornchai, 1982; Mosier, 1985; Chu, 1989; Taboun *et al.*, 1991; Kusiak, 1992; Heragu, 1994; Liang and Taboun, 1995; Joines *et al.*, 1996; Selim *et al.*, 1998; Abduelmula *et al.*, 1997; Abduelmula *et al.*, 1998; Taboun *et al.*, 1998a and 1998b). The majority of these studies assumed that each part has a single process plan, which reduces the system productivity (Adil *et al.*, 1996). In modern manufacturing systems, however, alternative process plans are available for parts, which in turn improve the group-ability of the part-machine matrix, that is producing manufacturing cells with a minimum number of inter-cell movements (Kusiak and Cho, 1992). Parts can also be re-scheduled to alternative process plans in case of abnormal situations such as machine breakdowns, and capacity problems. For instance, consider a 4 parts by 5 machines matrix presented in Table 5.1. The final solution of this matrix results in two manufacturing cells (Table 5.2). Suppose that parts 2 and 3 in Table 5.1 have alternative process plans that involve machines 2 and 3 for part 2, and machines 4 and 5 for part 3. This will reduce the number of inter-cell movements from 2 to 0. Therefore, it is important to consider different process plans when designing a CM system.

So far, very few models have been developed to solve the CF problem considering different process plans. Adil *et al.* (1996) and Srinivasan and Narendran (1991) described the intra and inter-cell handling movements in terms of number of voids and exceptional elements respectively. However, neither considers important factors such as production volume, selling price, and maximum number of machines in each cell due to space limitations.

Part	Machines				
	M1	M2	M3	M4	M5
P1	1	0	0	1	1
P2	0	1	0	1	1
P3	0	1	1	1	0
P4	0	1	1	0	0

Table 5.1. Original part-machine Matrix.

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*. Each part has one process plan.

Part	Machines				
	M1	M4	M5	M2	M3
P1	1	1	1	0	0
P2	0	1	1	1	0
P3	0	1	0	1	1
P4	0	0	0	1	1

Table 5.2. Part families and machine cells.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

The motivation for this chapter is to develop a productivity model to design a CM system considering alternative process plans. This model forms part families and machine groups simultaneously, and selects the process plan such that the system productivity is optimized. First, a 0-1 integer-programming model is developed to maximize system productivity defined as the ratio of output to the total material handling cost. Second, a simulated annealing algorithm is developed to solve large-scale problems. The following section presents the problem statement and assumptions. Section 5.3 explains the development of the mathematical model, while a simulated annealing algorithm is given in section 5.4. A comparison between the mathematical model and the simulated annealing model is presented in section 5.5. Section 5.6 presents a computational experience with large problems. Advantages and disadvantages of the proposed models are presented in section 5.7.

5.2. Problem Statement and Assumptions

The problem consists of a manufacturing system that involves several machines and a number of parts to be processed, each part needing a certain number of operations. Each part has alternative process plans. The objective is to organize these parts and machines into part families and machine groups and select the process plan such that the system productivity is optimized. It is assumed that the parts have a deterministic demand at the beginning of the planning period. It is also assumed that the maximum number of cells is equal to the number of machines in the problem and that the model determines the optimum number of cells. Therefore, initially each machine is assigned to a separate manufacturing cell: machine 1 to cell 1, machine 2 to cell 2, machine M to cell C.

5.3 Mathematical Model (MM)

In this section, a 0-1 integer-programming model is developed to maximize productivity, P where:

$$P=\frac{\sum_{i}\sum_{r}\sum_{k} D_i * S_i * Y_{irk}}{\sum_{i}\sum_{r}\sum_{k} NM_{ir} * IMC * D_i * Y_{irk} + \sum_{i}\sum_{r}\sum_{j}\sum_{k} (1-X_{jk}) * b_{ij} * E_{ir} * Y_{irk} * EMC * D_i} \qquad (5.1)$$

Subject to

$$\sum_{j} X_{jk} < M_{max} \qquad \forall k \qquad (5.2)$$

$$\sum_{k} X_{jk} = 1 \qquad \forall j \qquad (5.3)$$

$$\sum_{k}\sum_{r} Y_{irk} = 1 \qquad \forall i \qquad (5.4)$$

$$X_{jk} = 0 \text{ or } 1 \qquad \forall (j,k) \qquad (5.5)$$

$$Y_{irk} = 0 \text{ or } 1 \qquad \forall (i,r,k) \qquad (5.6)$$

The objective function for this model presented in equation (5.1), maximizes the total productivity defined as the output divided by input. The output is defined as the number of parts produced multiplied by their selling prices. The input has two non-linear terms. The first term captures the intra-cell handling cost. The second term captures the inter-cell handling cost. The maximum number of machines in each cell is determined by constraint (5.2). Constraint (5.3) states that each machine is assigned to only one cell.

Constraint (5.4) states that in the final assignment, each part family is assigned to only one cell. The integrity of the decision variables is ensured by constraints (5.5) and (5.6).

5.4 Illustrative Examples

In this section, four problems of different size and structure are considered to illustrate the application of the developed model. Information regarding annual demand is randomly generated between 200 and 300 units (Appendix 1). The selling price of each part is also randomly generated between \$10 and \$15 (Appendix 1). The intra and inter-cell movement costs are assumed to be \$2 and \$4.6 per movement respectively in these problems. These problems are solved using LINGO software on a Pentium 120 MHz computer. The input and output files for problem 1 are selected as an example and presented in Appendix 3.

5.4.1 Example Problem 1

This problem was considered by Kusiak (1987) and Adil *et al.* (1996). Information regarding the original part-machine matrix is presented in Table 5.3. The demand and selling prices are presented in Table 5.4. The maximum number of machines per cell is assumed to be 2. Table 5.5 presents the final manufacturing cells and optimum process plans.

5.4.2 Example Problem 2

A 5 parts by 4 machines problem considered by Adil *et al.* (1996) is selected for this example. The original part-machine matrix is presented in Table 5.6. Demand and selling price are presented in Table 5.7. The maximum number of machines is assumed to be 2. Table 5.8 presents the final part families, machine cells and the optimum process plan.

5.4.4 Example Problem 3

A 10 parts by 8 machines problem is generated from problem 2 (Table 5.6) by replicating each row and each column two times. The original part-machine matrix is presented in Table 5.9. Generated demand and selling price are presented in Table 5.10. The maximum number of machines per cell is assumed to be 4. Table 5.11 presents the final part families, machine cells, and optimum process plan.

5.4.4 Example Problem 4

In this example, a problem of 15 parts by 10 machines generated from problem 2 (Table 5.6) is used. The part-machine relationship is presented in Table 5.12. Information about demand and selling prices is presented in Table 5.13. The maximum number of machines per cell is assumed to be 4. Table 5.14 presents the final part families and machine groups.

Part No	Process Plan	Machine 1	Machine 2	Machine 3	Machine 4
1	1	0	0	1	1
	2	0	1	0	1
	3	1	1	0	0
2	1	0	1	1	0
	2	1	0	1	0
3	1	1	0	0	1
	2	0	1	0	1
4	1	1	0	0	1
	2	1	0	1	0
5	1	0	0	1	1
	2	1	0	0	0

Table 5.3. Machine-part matrix from problem 1.

Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

Part	1	2	3	4	5
D	239	210	203	248	260
S	10	10	14	12	13

Table 5.4. Generated demand and selling price for problem 1.

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between $10 and $15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Objective function = 3.3223		CPU Time=12 (sec)
Cell	Part (process plan)	Machines
1	1(2),3(2)	2,4
2	2(2),4(2),5(2)	1,3

Part Type (Process Plan)	Machine Number 2	4	1	3
1(1)	1	1		
3(2)	1	1		
2(2)			1	1
4(2)			1	1
5(2)			1	

Table 5.5. The mathematical programming solution for problem 1.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part No	Process Plan	Machine			
		1	2	3	4
	1	0	0	1	1
1	2	0	1	0	1
	3	1	1	0	0
2	1	0	1	1	0
	2	1	0	1	0
3	1	1	0	0	1
	2	0	1	0	1
4	1	1	0	0	1
	2	1	0	1	0
	1	0	0	1	1
5	2	1	0	0	0
	1	1	1	1	0

Table 5.6. Machine-part relationship from problem 2.

Note: In this matrix, the element 0 means that part type p does not require machine type m, while the element 1 means that part type p requires machine type m.

Part	1	2	3	4	5
D	239	210	203	248	260
S	10	10	14	12	13

Table 5.7. Generated demand and selling price for problem 2.

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between $10 and $15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Objective function = 3.3223		CPU Time=16 (sec)
Cell	Part (process plan)	Machines
1	1(2),3(2)	2,4
2	2(2),4(2),5(2)	1,3

Part Type (Process Plan)	Machine Number 2	4	1	3
1(2)	1	1		
3(2)	1	1		
2(2)			1	1
4(2)			1	1
5(2)			1	

Table 5.8. The mathematical programming solution for problem 2.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part No	Process Plan	Machine 1	2	3	4	5	6	7	8
1	1	0	0	1	1	0	0	1	1
	2	0	1	0	1	0	1	0	1
	3	1	1	0	0	1	1	0	0
2	1	0	1	1	0	0	1	1	0
	2	1	0	1	0	1	0	1	0
3	1	1	0	0	1	1	0	0	1
	2	0	1	0	1	0	1	0	1
4	1	1	0	0	1	1	0	0	1
	2	1	0	1	0	1	0	1	0
5	1	0	0	1	1	0	0	1	1
	2	1	0	0	0	1	0	0	0
	3	1	1	0	0	1	1	0	0
6	1	0	0	1	1	0	0	1	1
	2	0	1	0	1	0	1	0	1
	3	1	1	0	0	1	1	0	0
7	1	0	1	1	0	0	1	1	0
	2	1	0	1	0	1	0	1	0
8	1	1	0	0	1	1	0	0	1
	2	0	1	0	1	0	1	0	1
9	1	1	0	0	1	1	0	0	1
	2	1	0	1	0	1	0	1	0
10	1	0	0	1	1	0	0	1	1
	2	1	0	0	0	1	0	0	0
	3	1	1	0	0	1	1	0	0

Table 5.9. Machine-part relationship from problem 3.

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

Part	1	2	3	4	5	6	7	8	9	10
D	239	210	203	248	260	239	210	203	248	260
S	10	10	14	12	13	10	10	14	12	13

Table 5.10. Generated demand and selling price for problem 3.

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between \$10 and \$15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Objective function = 1.0991	CPU Time=600 (sec)	
Cell	Part (process plan)	Machines
1	1(3),3(1),4(1),5(2), 8(1),9(1),10(2)	2,4,6,8
2	7(1)	3,7
3	10(2),5(2)	1,5

Part Type (Process Plan)	Machine Number 2	4	6	8	3	7	1	5
1(2)	1	1	1	1				
2(1)	1		1		1	1		
3(2)	1	1	1	1				
4(1)		1		1			1	1
6(2)	1	1	1	1				
8(2)	1	1	1	1				
9(1)		1		1			1	1
7(1)	1		1		1	1		
5(2)							1	1
10(2)							1	1

Table 5.11. The mathematical programming solution from problem 3.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Part No	Process Plan	Machine									
		1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	1	0	0	1	1	0	0
	2	0	1	0	1	0	1	0	1	0	1
	3	1	1	0	0	1	1	0	0	1	1
2	1	0	1	1	0	0	1	1	0	0	1
	2	1	0	1	0	1	0	1	0	1	0
3	1	1	0	0	1	1	0	0	1	1	0
	2	0	1	0	1	0	1	0	1	0	1
4	1	1	0	0	1	1	0	0	1	1	0
	2	1	0	1	0	1	0	1	0	1	0
5	1	0	0	1	1	0	0	1	1	0	0
	2	1	0	0	0	1	0	0	0	1	0
	3	1	1	0	0	1	1	0	0	1	1
6	1	0	0	1	1	0	0	1	1	0	0
	2	0	1	0	1	0	1	0	1	0	1
	3	1	1	0	0	1	1	0	0	1	1
7	1	0	1	1	0	0	1	1	0	0	1
	2	1	0	1	0	1	0	1	0	1	0
8	1	1	0	0	1	1	0	0	1	1	0
	2	0	1	0	1	0	1	0	1	0	1
9	1	1	0	0	1	1	0	0	1	1	0
	2	1	0	1	0	1	0	1	0	1	0
10	1	0	0	1	1	0	0	1	1	0	0
	2	1	0	0	0	1	0	0	0	1	0
	3	1	1	0	0	1	1	0	0	1	1
11	1	0	0	1	1	0	0	1	1	0	0
	2	0	1	0	1	0	1	0	1	0	1
	3	1	1	0	0	1	1	0	0	1	1
12	1	0	1	1	0	0	1	1	0	0	1
	2	1	0	1	0	1	0	1	0	1	0
13	1	1	0	0	1	1	0	0	1	1	0
	2	0	1	0	1	0	1	0	1	0	1
14	1	1	0	0	1	1	0	0	1	1	0
	2	1	0	1	0	1	0	1	0	1	0

Table 5.12. Machine-part relationship from problem 4.

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

15	1	0	0	1	1	0	0	1	1	0	0
	2	1	0	0	0	1	0	0	0	1	0
	3	1	1	0	0	1	1	0	0	1	1

Table 5.12 (Cont.). Machine-part relationship from problem 4.

Note: In this matrix, the element 0 means that part type *p* does not require machine type *m*, while the element 1 means that part type *p* requires machine type *m*.

Part	1	2	3	4	5	6	7	8	9	10
D	299	291	239	210	203	281	248	260	273	299
S	14	14	11	10	10	14	12	13	13	14
Part	11	12	13	14	15					
D	239	210	203	281	248					
S	11	10	10	14	12					

Table 5.13. Generated demand and selling price for problem 4.

Note: Information regarding annual demand (D) and selling price (S) is generated randomly between 200 and 300 units and between $10 and $15 respectively. This was done using a FORTRAN subroutine adapted from Microsoft FORTRAN presented in Appendix 1.

Objective function = 0.9270		CPU Time=3617(sec)
Cell	**Part (process plan)**	**Machines**
1	1(1),6(1),11(1)	4,7,8,10
2	2(2),3(1),4(2),5(2), 7(2), 8(1), 9(2), 10(2), 12(2), 13(1) , 14(2), 15(2)	1,3,5,9
3		2,4

Part Type (Process Plan)	4	7	8	10	1	3	5	9	2	6
1(1)	1	1	1			1				
6(1)	1	1	1			1				
11(1)	1	1	1			1				
2(2)		1			1	1	1	1		
3(1)	1		1				1	1		
4(2)	1	1			1		1	1		
5(2)					1		1	1		
7(2)		1			1	1	1	1		
8(1)	1		1		1		1	1		
9(1)		1			1	1	1	1		
10(2)					1		1	1		
12(2)		1			1	1	1	1		
13(1)	1		1		1		1	1		
14(2)		1			1	1	1	1		
15(2)					1		1	1		

Table 5.14. The mathematical programming solution for problem 4.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

5.5 Simulated Annealing Algorithm

In this section, the simulated annealing algorithm developed in chapter 4 is extended to solve the cell formation problem in CM systems considering alternative process plans. A summary of the proposed algorithm is presented in Figure 5.1. The proposed algorithm is coded in FORTRAN 77 using a Pentium 120 MHz computer. The FORTRAN program of this algorithm is presented in Appendix 4. The following section explains how the proposed SA algorithm works considering alternative process plans.

1. Input

   1.1. Read the input data: number of parts P, number of machines M, number of cells C, process plans PP, demand D, selling price S, inter-cell handling cost EMC, intra-cell handling cost IMC, and maximum number of machines allowed in each cell M_{max}.

   1.2. Define the annealing parameters: initial temperature T_0, final temperature T_f, decrement factor α, and the number of iterations at each temperature level NIRAT, where NIRAT= K*NM, where K is a constant, called the size factor that controls the cooling rate, and NM is the number of machines in the problem.

2. Initialization

   2.3. Set the counter i = 0

   2.4. Find an initial solution. Calculate the total material handling cost by assuming that each part is moving through all cells to complete the required operations. Then calculate the initial objective function OBJ_0 of the current manufacturing cells.

2.3. Let the objective function OBJ_0=OBJ=BOBJ, where OBJ is the current objective function and BOBJ is the best objective function found so far.

3. Set the number of iterations at each temperature level IR=0

4. Generate a neighboring solution

 4.1. Set IR=IR+1

 4.2. Randomly select a machine to move to a randomly selected cell and re-calculate the objective function of the current solution. Set the new solution = NOBJ

5. Evaluation

 5.1. If NOBJ $\geq$ OBJ, then set OBJ=NOBJ, and go to step 6

 5.2. If NOBJ $\geq$ BOBJ, then set BOBJ=NOBJ, and go to step 6

 5.3. If NOBJ< OBJ, then let the probability of acceptance of a new solution equal $e^{(-\Delta z/T)}$, where Δz = NOBJ-OBJ. Choose a random number R in (0,1); if R $\leq e^{(-\Delta z/T)}$, then set OBJ=NOBJ. Otherwise reject the solution and go to step 6.

6. If the number of iterations IR<NIRAT, then go to 4.1. Otherwise go to step 7.

7. Adjust the temperature, set the current temperature $T = \alpha T$ and go to step 8

8. If $T \leq T_f$, then go to step 9. Otherwise go to step 3.

10. Part assignment

 Calculate the number of machines required by each part in all cells and assign the part to the cell with the maximum number of machines and go to step 10.

11. Print the best solution obtained and stop.

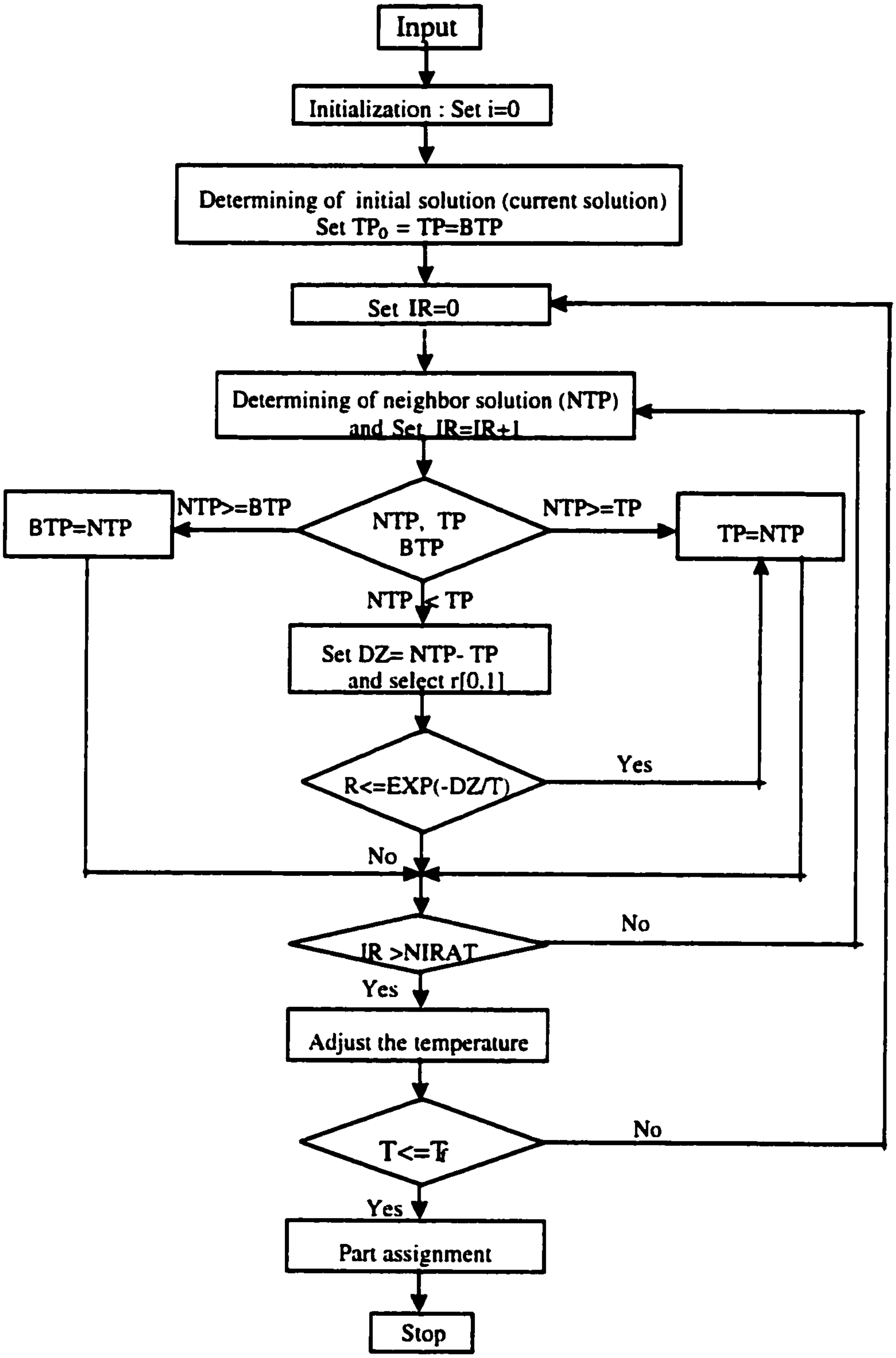


Figure 5.1. Flow chart of the SA algorithm.

5.6 Numerical Examples

In this section, the four problems used in section 5.2 are used to test the capability of the developed simulated annealing algorithm. The performance of simulated annealing depends on the initial temperature, cooling rate, and final temperature. Therefore, the experiments conducted in section 4.4 are used in this section to determine the values of the above parameters.

Table 5.15 presents the effect of initial temperature T_0 on both objective function and CPU time. It is clear that for problems 1 and 2 the initial temperature $T_0 = 0.1$ has the best objective function and shortest CPU time; therefore, it is selected as the best T_0 for both problems. Clearly, for problem 3, the most promising initial temperatures are $T_0 = 50$, $T_0 = 10$, and $T_0 = 5$. Both $T_0 = 50$ and $T_0 = 10$ are eliminated because of their long computational times; therefore, $T_0 = 5$ is selected for problem 3. Similarly, the best initial temperature for problem 4 is $T_0 = 0.5$.

The cooling rate is controlled by the size factor and decreasing factor. The effect of the size factor k is studied first and the results are presented in Table 5.16. The size factor k = 1 results in highest productivity value and shortest computational time for problems 1, 2, 3 and 4. Therefore, it is selected as the best k for all problems

Table 5.17 presents the effect of the decreasing factor α on both productivity and CPU time. For problems 1 and 2, α=0.96 is selected since it has the highest productivity value and shortest computational time. For problems 2 and 3, α=0.99 provides the best values in both productivity and computational time.

The effect of the stopping temperature T_f on both productivity and running time is presented in Table 5.18. Clearly, the most promising stopping temperature is T_f =0.05 for all problems. Therefore, it is selected as the best T_f for all problems. Clearly, no standard annealing parameters can be recommended. The results obtained using the parameters for all problems are summarized in Tables 5.19 to 5.22.

T_0		50	10	5	1	0.5	0.1
P1	IPA	0.99	0.95	0.92	0.66	0.45	0.30
	CPU	0.49	0.33	0.27	0.22	0.11	0.06
	OBJ	3.3223	3.3223	3.3223	3.3223	3.3223	3.3223
P2	IPA	0.99	0.96	0.93	0.70	0.41	0.21
	CPU	0.33	0.22	0.17	0.11	0.8	0.5
	OBJ	3.3223	3.3223	3.3223	3.3223	3.3223	3.3223
P3	IPA	0.99	0.98	0.97	0.90	0.81	0.35
	CPU	3.51	2.69	2.31	1.49	1.15	0.33
	OBJ	1.3231	1.3231	1.3231	1.2214	1.2214	1.1059
P4	IPA	0.99	0.99	0.99	0.95	0.91	065
	CPU	9.67	7.85	6.43	4.23	3.24	0.99
	OBJ	1.0767	1.0767	1.0767	1.0767	1.0767	0.8627

Table 5.15. Selection of initial temperature (T_0) for all problems.

Note: IPA = initial probabability of acceptance; CPU = computional time in seconds; OBJ = objective function.

K		1	2	4	8	16	32
P1	CPU	0.06	0.08	0.11	0.22	0.50	0.93
	OBJ	3.3223	3.3223	3.3223	3.3223	3.3223	3.3223
P2	CPU	0.05	0.11	0.19	0.28	0.55	1.05
	OBJ	3.3223	3.3223	3.3223	3.3223	3.3223	3.3223
P3	CPU	2.31	4.61	9.28	18.78	37.35	1.15.13
	OBJ	1.3231	1.3231	1.3325	1.3325	1.3325	1.3325
P4	CPU	3.24	6.48	12.90	25.76	53.15	1.56.10
	OBJ	1.0767	1.0767	1.0767	1.0767	1.0767	1.0767

Table 5.16. Selection of the size factor (K) for all problems.

Note: CPU = computational time in seconds; OBJ = objective function.

α		0.99	0.98	0.96	0.92	0.85
P1	CPU	0.06	0.05	0.04	0.03	0.03
	OBJ	3.3223	3.3223	3.3223	2.2249	2.2249
P2	CPU	0.05	0.05	0.04	0.04	0.04
	OBJ	3.3223	3.3223	3.3223	2.2249	2.2249
P3	CPU	2.31	1.54	0.60	0.28	0.16
	OBJ	1.3231	1.2214	1.2214	1.1059	1.1059
P4	CPU	3.24	1.59	0.82	0.60	0.22
	OBJ	1.0767	0.8627	0.8627	0.8627	0.8627

Table 5.17. Selection of the decreasing factor (α) for all problems.

Note: IPA= CPU = computational time in seconds; OBJ = objective function.

T_f		0.1	0.05	0.01	0.005	0.001
P1	CPU	0.03	0.04	0.11	0.16	0.22
	OBJ	2.1988	3.3223	3.3223	3.3223	3.3223
P2	CPU	0.03	0.04	0.11	0.17	0.22
	OBJ	2.1988	3.3223	3.3223	3.3223	3.3223
P3	CPU	1.89	2.31	3.13	3.90	4.73
	OBJ	1.3231	1.3231	1.3231	1.3231	1.3231
P4	CPU	2.26	3.24	5.49	6.48	8.74
	OBJ	0.8627	1.0767	1.0767	1.0767	1.0767

Table 5.18. Selection of stopping conditions (T_f) for all problems.

Note: IPA= CPU = computational time in seconds; OBJ = objective function.

Objective function = 3.3223	CPU Time=0.04 (sec)	
Cell	Part (process plan)	Machines
1	1(2),3(2)	2,4
2	2(2),4(2),5(2)	1,3

Part Type (Process Plan)	Machine Number 2	4	1	3
1(1)	1	1		
3(2)	1	1		
2(2)			1	1
4(2)			1	1
5(2)			1	

Table 5.19. The simulated annealing solution for problem 1.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Objective function = 3.3223	CPU Time=0.04 (sec)	
Cell	Part (process plan)	Machines
1	1(2),3(2)	2,4
2	2(2),4(2),5(2)	1,3

Part Type (Process Plan)	Machine Number 2	4	1	3
1(2)	1	1		
3(2)	1	1		
2(2)			1	1
4(2)			1	1
5(2)			1	

Table 5.20. The simulated annealing solution for problem 2.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Objective function = 1.3231		CPU Time=2.31 (sec)
Cell	Part (process plan)	Machines
1	1(2),2(1),3(2),4(1), 6(20),7(1),8(2),9(1),	2,4,6,8
2		3,7
3	5(2),10(2))	1,5

Part Type (Process Plan)	Machine Number 2	4	6	8	3	7	1	5
1(2)	1	1	1	1				
2(1)	1		1		1	1		
3(2)	1	1	1	1				
4(1)		1		1			1	1
6(2)	1	1	1	1				
8(2)	1	1	1	1				
9(1)		1		1			1	1
7(1)	1		1		1	1		
5(2)							1	1
10(2)							1	1

Table 5.21. The simulated annealing solution for problem 3.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

Objective function = 1.0767		CPU Time=64(sec)
Cell	Part (process plan)	Machines
2	2(2),3(1),4(2),5(2), 7(2),8(1),9(2),10(2), 12(2),13(1),14(2), 15(2)	1,5,9
2	1(1),6(1),11(1)	3,4,7,8
3		10
4		6
5		2

Part Type (Process Plan)	Machine Number 3	4	7	8	1	5	9	2	6	10
1(1)	1	1	1	1						
6(1)	1	1	1	1						
11(1)	1	1	1	1						
2(2)	1		1		1	1	1			
3(1)		1		1		1	1			
4(2)		1	1		1	1	1			
5(2)					1	1	1			
7(2)	1		1		1	1	1			
8(1)		1		1	1	1	1			
9(1)	1		1		1	1	1			
10(2)					1	1	1			
12(2)	1		1		1	1	1			
13(1)		1		1	1	1	1			
14(2)	1		1		1	1	1			
15(2)					1	1	1			

Table 5.22. The simulated annealing solution for problem 4.

Note: This table represents a solution matrix. Each block in the matrix represents a manufacturing cell with its part family and machine group.

5.7 Comparison of the Algorithms

In this section, the four problems presented in sections 5.4 and 5.6 are used to compare the performance of the mathematical model (MM) with the simulated annealing (SA) algorithm. The results of applying both models to the four problems are summarized in Table 5.23. The following paragraphs present a brief analysis for the four problems.

Problems 1 and 2: It is clear that both models provide the same productivity values. However, the SA took less computational time to find the optimum manufacturing cells. Both models form two manufacturing cells with the same cell configurations.

Problem 3: The SA outperforms the MM in both productivity and computational time. The productivity value obtained by the SA is 20% higher than the value obtained by the MM. Both models form 3 manufacturing cells with different cell configurations.

Problem 4: There is a significant difference between the models in solving this problem, especially in the computational time. The MM spends 3617 sec while the SA spends only 64 sec to solve the problem. The SA also outperforms the MM in productivity value. The productivity value obtained by the SA is 16% higher than the value obtained by the MM.

Problem Number.	CPU (sec)		Productivity	
	MM	SA	MM	SA
P1	12	1	3.3223	3.3223
P2	16	1	3.3223	3.3223
P3	600	2.31	1.0991	1.3231
P4	3617	64	0.9270	1.0767

Table 5.23. A comparison of SA with MM.

5.8 Computational Experience with Large Problems

The performance of the developed SA algorithm for large-scale problems is discussed in this section. Two example problems are generated from problem 2 (Table 5.6) by replicating the rows and columns of the original part-machine matrix. For example, a 30 part and 16 machine problem is generated by replicating each row six times and each column four times. Information regarding annual demand is generated between 200 and 300 units for all problems, and the selling price of each part is generated between \$10 and \$15. The intra and inter-cell movement costs are assumed to be \$2 and \$4.6 per movement respectively, in all problems.

The annealing parameters are assumed as follows: $T_0 = 10$; $T_f = 0.05$; $\alpha = 0.99$ and K=4. All problems are solved using a Pentium 120 MHz computer. Table 5.24 presents a summary of the computational results for all problems. The results show the ability of the developed SA algorithm to solve large-scale problems.

Problem number	Parts	Machines	M_{max}	Objective function	Computational time (sec)
1	30	16	5	0.4908	207
2	100	32	7	1.9902	6960

Table 5.24. Performance of the developed SA algorithm for large-scale problems.

5.9 Advantages and Disadvantages of the Developed Productivity Model

This section presents the advantage and disadvantages of the developed productivity models. As illustrated in the previous chapter, both the mathematical and simulated annealing (SA) models have several advantages and disadvantages.

5.9.1 Advantages

The advantages of the developed models can be summarized as follows:

1. The developed models use a direct index "productivity index" to form part families and machine groups.
2. The developed models are easy to understand and use by practitioners since the only input data are the number of parts, machines and cells, process plans, demand, selling price, intra and inter-cell costs and maximum number of machines allowed in each cell.
3. The developed models incorporate different process plans for handling parts in case of abnormal situations such as machine breakdowns, and capacity problems.
4. The developed models form part families and machine groups simultaneously. Thus, no additional procedure is needed to solve the problem.
5. The developed models do not require an advance determination of the number of manufacturing cells.
6. The developed models have the flexibility to control the minimum number of cells by considering the maximum number of machines in each cell due to space and safety limitations.
7. The developed SA model provides the type of parts and machines in each manufacturing cell. Thus, the final solution does not require a visual investigation to identify part families and machine cells.
8. The developed SA model has the ability to solve large-scale problems in a reasonable amount of time.

5.9.2 Disadvantages

The developed algorithms also have the following disadvantages:

1. The MM model requires visual inspection of the final solution to identify part families and machine groups.
2. The MM model requires long computational times to solve large-scale problems.
3. Although the developed models consider some important factors such as production volume, total material handling cost, and maximum number of machines in each cell, they do not consider other factors such as machine capacity, sequence of operations, safety requirements, and machine reliability.
4. The developed models also do not consider other input factors such as labour, tooling, fixturing, machine investment, and maintenance costs.

CHAPTER 6

CONCLUSIONS AND FURTHER WORK

In this chapter, conclusions regarding cellular manufacturing design are discussed in section 6.1, contributions of the research are presented in section 6.2, and directions for further research are presented in section 6.3.

6.1 Conclusions

The need for higher productivity is forcing traditional manufacturers to consider changing and improving their facilities. Traditional manufacturing systems are classified into job shop, batch, and mass production systems. These manufacturing systems have to eliminate some disadvantages of their facilities such as long setup time, long waiting time, and higher material movement to achieve higher productivity. Cellular Manufacturing (CM) systems have been approved to overcome these drawbacks. The main aim of a CM system is to solve the cell formation problem in which similar parts are grouped in part families and their required machines into machine cells.

Many algorithms have been developed to design CM Systems over the last two decades. A comprehensive literature review reveals that the majority of those models have the following characteristics:

- They use indirect measures such as similarity or dissimilarity coefficient methods to form part families and machine cells.
- They require visual inspection of the solution to identify part families and machine cells.

- They do not take into consideration some important factors that affect the final solution such as production volume.
- The number of manufacturing cells must be specified in advance.
- They do not simultaneously form the part families and machine cells.
- They require long computational times to solve large-scale problems.

In this research, a solution for the grouping problem in CM systems is proposed with the objective to optimize system productivity to overcome the above limitations. In chapter 3, a cellular manufacturing design problem is considered with the objective of maximizing the system productivity. The problem is formulated as a 0-1 integer-programming model subject to a maximum number of machines per cell constraint, and each machine is assigned to only one cell. The developed model offers some advantages such as the ability to form part families and machine groups simultaneously, and considers some important factors such as production quantity and selling price. Furthermore, in this model the number of manufacturing cells does not need to be specified in advance. Problems of different size and complexity selected from the literature are used to test the developed model, and the results show the ability of the developed model to determine the optimum solutions for all problems. However, long computational times are required to get the optimum solution for large size problems. Hence, a simulated annealing algorithm has been presented in chapter 4 to solve large-size problems. The algorithm is applied to the same problems, and the results show the superiority of SA algorithm over the mathematical programming model in both objective function and computational time. The ability of the developed SA algorithm to solve a large-scale problem is also discussed. A detailed analysis was conducted for different problems to determine suitable values for the annealing parameters.

In chapter 5, the total productivity model developed in chapter 3 is extended to consider alternative process plans. First, a non-linear 0-1 integer-programming model is developed to maximize the total productivity defined as the ratio of output to the total material handling cost. Second, a simulated annealing algorithm is developed to solve large-scale problems. An experimental analysis is conducted to select the annealing parameters.

6.2 Contributions of the Research

This thesis contributes to the area of cellular manufacturing (CM) systems by introducing a productivity model that forms part families and machine groups simultaneously and selects the process plan such that the system productivity is optimized. The contribution of this research is summarized as follows:

1. Developing a 0-1 integer-programming model to design a CM system by using a direct productivity index. The developed model takes into consideration the following :
   - Intra and inter-cell handling costs
   - Selling price
   - Production volume
   - Maximum number of machines in each cell
2. Developing a simulated annealing algorithm to solve large-scale problems taking into consideration the above factors.
3. Conducting a detailed analysis on different problems to test the effect of various

annealing parameters on the solution quality.

4. Extending both 0-1 integer programming and simulated annealing to consider alternative process plans.

6.3 Further Research

The following are some potential directions for further research:

- In this research, we considered some important factors such as selling price, demand and the maximum number of machines in each manufacturing cell. This can be extended further to consider other technological factors such as machine capacity, sequence of operations, safety requirements and machine reliability.
- The developed model considers only the total material handling cost input. It would be interesting to consider other input factors such as labour, tooling, fixturing, and machine investment and maintenance costs.
- Another interesting direction for future research would be to use other search techniques such as genetic algorithms and tabu search to solve the developed model and compare the results with those obtained by using the SA algorithm.

REFERENCES

Abduelmola, A. I., Taboun, S. M., and Koa, D., "Manufacturing cells formation with productivity consideration". *Proceedings of the 2nd Annual International Conference on Industrial Engineering Applications and Practice,* San Diego, California, U. S. A., November 1997a, pp. 779-784.

Abduelmola, A. I., Mohamed, S. M., and Koa, D., "Productivity modeling of manufacturing facilities: a case study". *Proceedings of the 2nd Annual International Conference on Industrial Engineering Applications and Practice,* San Diego, California, U. S. A., November, 1997b, pp. 971-976.

Abduelmola, A. I., Taboun, S. M., and Merchawi, S., "Productivity Optimization of Cellular Manufacturing Systems". Computers and Industrial Engineering. 1998, Vol. 35, No. 3-4, 403-406.

Adil, G. K., Rajamani, D. and Strong D., "Cell Formation Considering Alternative Routings". International Journal of Production Research. 1996, Vol. 34, No. 5, 1361-1381.

Amira, F., and Choobineh F., "Identifying the Composition of Cellular Manufacturing Systems". International Journal of Production Research. 1996, Vol. 34, No. 9, 2471-2488.

Askin, R. G., and Chiu, K. S., "A Graph Partitioning Procedure for Machine Assignment and Cell Formation in Group Technology". International Journal of Operational Research. 1990, Vol. 28, N0, 8, 1555-1572.

Askin, R. G., and Subramanian, S. P., "A Cost-Based Heuristic for Group Technology Configuration". International Journal of Production Research. 1987, Vol. 25, No. 1, 101-113.

Ballakur, A., and Steudel, H. J., "Within Cell Utilization Based Heuristic for Designing Cellular Manufacturing Systems". International Journal of Production Research. 1987, Vol. 25, 639-665.

Beaulieu, A., Gharbi, A., and Ait-Kadi, "An Algorithm for the Cell Formation and the Machine Selection Problems in the Design of Cellular Manufacturing Systems". International Journal of Production Research. 1997, Vol. 35, No.7, 1857- 1874.

Boctor, F. I., "Minimum-Cost Machine -Part Cell Formulation Problem". International Journal of Production Research. 1996, Vol. 34, 1045-1063.

Boe, W. J., and Cheng, C. H., "A Close Neighbor Algorithm for Designing Cellular

Manufacturing Systems". International Journal of Production Research. 1991, Vol. 29, No 10, 2097-2116.

Brown, J., John H., and James, S., "Production Management Systems: a CIM Perspective". Addison-Wesley Publishing Company, Reading MA, 1988.

Burbidge, J. L., "Production Flow Analysis". Production Engineer. December, 1963, Vol. 42, No. 12, 742-752.

Burbidge, J. L., and Halsall, J., "Group Technology and Growth at Shalibane". Production Planning & Control. 1994, Vol. 5, No. 2, 213-218.

Craig, C. E., and Harris, C. R, "Total Productivity Measurement at the Firm Level". Sloan Management Review. 1973, Vol. 14, 13-29.

Carrie, S. A., "Numerical Taxonomy Applied to Group Technology and Plant Layout". International Journal of Production Research. 1973, Vol. 11, No. 4, 399-416.

Chan, H. M., and Milner, D. A., "Direct Clustering Algorithm for Group Formation in Cellular Manufacturing". Journal of Manufacturing Systems. 1982, Vol. 1, No. 1, 65-74.

Chandrasekharan, M. P., and Rajagopalan, R., "An Ideal Seed Nonhierarchical Clustering Algorithm for Cellular Manufacturing". International Journal of Production Research. 1986, No. 24, 451-464.

Chen, J. S., and Heragu, S, S., "Stepwise Composition Approaches for Large Scale Cell Formation Problems". European Journal of Operational Research. 1999, Vol. 113, No. 1, 64-79.

Chen, W., and Srivastava, B., "Simulated Annealing Procedures for Forming Machine Cells in Group Technology". European Journal of Operational Research. 1994, Vol. 75, No. 3, 100-111.

Choobineh, F. A., "Framework for the Design of Cellular Manufacturing Systems". International Journal of Operation Research. 1988, Vol. 26, NO. 7, 1161-1172.

Chow, W. S., and Hawaleshka, O., "A Novel Machine Grouping and Knowledge-Based Approach for Cellular Manufacturing". European Journal of Operational Research. 1993, 69, 357-372.

Chu, C. H., "Cluster Analysis in Manufacturing Cellular Formation". International Journal of Management Science. 1989, Vol. 17, No. 3, 289-295.

Cotruvo, N., "A Simulated Annealing Solution to the Cell Formation Problem". M.S. Thesis. 1993. Mississippi Stae University, USA.

Dahel, N. E., "Design of Cellular Manufacturing Systems in Tandem Configuration". International Journal of Production Research. 1995, Vol. 33, No. 8, 2079-2095.

Damodaran, V., Lashkari, R. S., and Singh, N., A "Production Planning Model for Cellular Manufacturing Systems with Refixturing Considerations". International Journal of Production Research. 1992, Vol. 30, No. 7, 1603-1615.

Dige, P., Lund, C., and Ravn, H., F., "Time Tabling by Simulated Annealing". Lecture Notes in Economics and Mathematical Systems: Applied Simulated Annealing. 1993, pp. 151-174.

Escoto, R. P., McDonnell, R. L., Salort, E. V., and Esteban F. C., "Development of an Algorithm for the Application of Group Technology in the Design of Manufacturing Systems". Production Planning & Control. 1998, Vol. 9, No.3, 267- 274.

Geonwook, J., Herman, R., L., and Parsaei, H., R., "A Cellular Manufacturing System Based on New Similarity Coefficient which Considers Alternative Routes During Machine Failure". International Journal of Production Research. 1998, Vol. 34, No.1, 21- 36.

Gunasingh, K. R., and Lashkari, R. S., "The Cell Formation Problem in Cellular Manufacturing System: a Sequential Modelling Approach". Computers in Engineering. 1989, Vol. 16, No. 4, 469-476.

Gupta, T., and Seifoddini, H., "Production Data Based Similarity Coefficient for Machine-Component Grouping Decisions in the Design of Cellular Manufacturing Systems". International Journal of Production Research. 1990, Vol. 28, No. 7, 1247-1269.

Ham, I., Hitomi, K., and Yoshiba, T., (1985), "Group Technology: Applications to Production Management". Kluwer-Nijhoff Publishing, Boston-Dordrecht-lancaster.

Han, G., and Ham. I., "Multiobjective Cluster Analysis for Part Family Formation". Journal of Manufacturing Systems. 1986, Vol. 5, No. 4, 223-230.

Harhalakis, G., Nagi, R., and Proth, J. M., "An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications". International journal of Production Research. 1990, Vol. 28, No. 1, 185-198.

Heragu, S. S., "Group Technology and Cellular Manufacturing". IEEE Transactions on Systems, Man, and Cybernetics. 1994, Vol. 24, No. 2, 203-215.

Heragu, S. S. and Gupta, Y. P., "A Heuristic for Design Cellular Manufacturing Facilities". International Journal of Production Research. 1994, Vol. 32, No. 1, 125-140.

Hitomi, K., "Manufacturing Systems Engineering". London, Taylor and Francis. 1979.

Husband, T.M., and Ghobadain, A., "Measuring Total Productivity in a Batch Factory: a Case Study". International Journal of Production Research. 1981, Vol. 19, pp. 411-424.

Hwang, H., and Ree P. "Routes Selection for the Cell Formation Problem with Alternative Part Process Plans". Computer and Industrial Engineering. 1996, Vol. 30, No. 3, 423-431.

Hwang, H., and Sun J., "A Genetic Algorithm-Based Heuristic for the GT Cell Formation Problem". Computer and Industrial Engineering. 1996, Vol. 30, No. 4, 941-955.

Jain, A, K., Kasilingam, R. G., and Bhole, S. D., "Joint Consideration of Cell Formation and Tool Provisioning Problem in Flexible Manufacturing Systems". Computer and Industrial Engineering. 1991, Vol. 20, No. 2, 271-277.

Jayakrishana, G., and Narendran, T., "CASE: a Clustering Algorithm for Cell Formation with Sequence Data". International Journal of Operation Research. 1998, Vol. 36, No. 1, 157-179.

Johnson, D., Aragon, C., McGeoch, L., and Schevan, C., "Optimization by Simulated Annealing and Experimental Evaluation Part I, Graph Partitioning", Operations Research. 1989, Vol. 37, No. 6, 865-892.

Joines, A. J., Culbreth, T. C., and King E. R., "Manufacturing Cell Design: an Integer Programming Model Employing Genetic Algorithms". IIE Transactions. 1996, 28, 69-85.

Kampersad, H. K., "Robotics Assembly System Design for Total Productivity". International Journal of Production Research. 1996, Vol. 34, No. 1, 71-93.

Kamrani, A. A., and Pawsaei H. R., "A Methodology for the Design of Manufacturing Systems Using Group Technology". Production Planning and Control. 1994, Vol. 5, No. 5, 450-464.

Kendrick, J. W., and Creamer, D., "Measuring Company Productivity: Hand Book With Case Studies". *Studies in Business Economics; National Industrial Conference Board,* New York, N. Y. 1969, No. 89.

Kendrick, J.W., "The Productivity Factor in Phase 2". The Conference Board Record, 1973, Vol. 9, No. 3, 28-35.

Kim, J., and Kim, Y., "Simulated Annealing and Genetic Algorithms for Scheduling Products with Multi-Level Product Structure". Computers and Operations Research. 1996, Vol. 23, No. 9, 857-868.

King, J. R., "Machine-Component Grouping in Production Flow Analysis: an Approach Using a Rank Order Clustering Algorithm". International Journal of Production Research. 1980, Vol. 18, No. 2, 213-232.

King, J. R. and Nakornchai, V., "Machine Component Grouping Formation in Group Technology: Review and Extension". International Journal of Production Research. 1982, Vol. 20, No. 2, 117-133.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing". Science. 1983, 220, 671-680.

Knauss, K., and Matuszal, M., "The Implementation of Cellular Manufacturing for US Factory Workers and their Unions". Labor Studies Journal. 1989, Vol. 14, 20-29.

Kusiak, A., "The Generalized Group Technology Concept". International Journal of Production Research. 1987, Vol. 25, No. 4, 561-569.

Kusiak, A., "Group Technology: Models and Solution Approaches". *First Industrial Engineering Research Conference Proceedings*. 1992, 349-352.

Kusiak, A. and Cho, M., "Similarity Coefficient Algorithms for Solving the Group Technology Problem". International Journal of Production Research. 1992, Vol. 30, No. 11, 2633-2646.

Laarhoven, P. J. M., and Aarts, E. H. L., "Simulated Annealing: Theory and Applications". Kuluwer Academic Publications, Dordrecht, Netherland. (1987)

Lee, H., and Garcia-Diaz, A., "A Network Flow Approach to Solve Clustering Problems in Group Technology". International Journal of Production Research. 1993, Vol. 31, No. 3, 603-612.

Lee, H., and Garcia-Diaz, A., "A Network Procedure for the Analysis of Cellular Manufacturing Systems". IIE Transactions. 1996, 28, 333-345.

Liang, M., and Taboun S. M., "Converting Functional Manufacturing Systems Focused Machine Cells: a Bicriterion Approach". International Journal of Production Research. 1995, Vol. 33, No. 8, 2147-2161.

Lin, L. T., Dessouky, M. M., Kumam, R. K., and Ng M. S., "A Heuristic-Based Procedure for The Weighted Production-Cell Formation Problem". IIE Transactions. 1996, 28, 579-589.

Logendran, R. A., "A Work Load Based Model for Minimizing Total Intercell and Intracell Moves in Cellular Manufacturing". International Journal of Production Research. 1990, Vol. 28, No. 5, 913-925.

Logendran, R. A., "Impact of Sequance of Operations and Layout of Cells in Celluar Manufacturing". International Journal of Production Research. 1991, Vol. 29, No. 2, 375-390.

Logendran, R., A "Binary Integer Programming Approach for Simultaneous Machine-Part Grouping in Cellular Manufacturing Systems". Computers and Industrial Engineering. 1993, Vol. 24, No. 3, 329-336.

May, J.D., and Denny, M., "Factor-Augmenting Technical Progress and Productivity in U.S. Manufacturing". Int. Economic Review. 1979, Vol. 20, No. 3, 759-773.

McAuley, J., "Machine Grouping for Efficient Production". The Production Engineer. Vol. 51, No. 2, 1972, 53-57.

Mohamed, A. M., "Productivity Models and Planning Models for Manufacturing Industries". Ph. D. Thesis. 1986, University of Manitoba, Canada.

Mosier, C. T., "The Facts of Group Technology and Their Impacts on Implementation: a State-of-the-survey". International Journal of Management Science. 1985, Vol. 13, No. 5, pp. 381-391.

Mukhopadhyay, S. K, Sarkar, P., Panda, R. P., "Machine-Component Grouping in Cellular Manufacturing by Multidimensional Scaling". International Journal of Production Research. 1994, Vol. 32, 457-477.

Mukhopadhyay, S. K, Singh, M. K., Srivstava, R., "FMS Machine Loading: a Simulated Annealing Approach". International Journal of Production Research. 1998, Vol. 36, No. 6, 1529-1547.

Okogba, G. O., Chen, M. T., Changchit, C. and Sell, R. L., "Manufacturing System Cell Formation and Evaluation Using a New Inter-cell Flow Reduction Heuristic". International Journal of Production Research. 1992, Vol. 30, 1101-1118.

Oliveira, J. F., and Ferreira, S. J., "Algorithms for Nesting Problems. Lecture Notes in Economics and Mathematical Systems: Applied Simulated Annealing". 1993, 255-273.

Rajagopalan, R., and Batra, J. L., "Design of Cellular Production Systems: A Graph Theoretic Approach". International Journal of Production Research. 1975, Vol. 13, No. 6, 567-579.

Rajamani, D., Singh, N., and Aneja, Y. P., "Integrated Design of Cellular Manufacturing Systems in the Presence of Alternative Process Plans". International Journal of Production Research. 1990, Vol. 28, No, 8, 1541-1554.

Rostogi, S. C., and Mohanty, R. P., "An Industrial Application for Total Productivity Optimization". Production Planning and Control. 1994, Vol. 5, No. 2, 219-228.

Rutenbar, R. A., "Simulated Annealing Algorithms: An Overview". IEEE. January1989, 19-26.

Sankaran, S., and Kasilingam, R. G., "On Cell Size and Machine Requirments Planning in Group Technology Systems". European Journal of Operational Research. 1993, Vol. 69, 373-383.

Sarker, B. R., and Balan, C. V., "Cell Formation with Operation Times of Jobs Even Distribution of Workloads". International Journal of Production Research. 1996, Vol. 34, No. 5, 1447-1468.

Seifoddini, H., "Duplication Process in Machine Cells Formation in Group Technology". IIE Transactions. 1989, 21, 382-388.

Seifoddini, H., and Djassemi, M., "A New Group Measure for Evaluation of Machine-Component Matrices". International Journal of Production Research . 1996, Vol. 34, No. 5, 1179-1193.

Selim, H. M., Askin, R. G., and Vakharia, A. J., "Cell Formation in Group Technology: Review, Evaluation and Direction for Future Research". Computer Industrial Engineering. 1998, Vol. 34, No, 1, 3-20.

Shafer, S. M., and Rogers, D. F., "Similarity and Distance Measures for Cellular Manufacturing. Part II. An Extension and Comparison". International Journal of Production Research. 1993, Vol. 31, No. 6, 1315-1326.

Shanker, K. S., and Srinivasulu, A., "Some Solution Methodologies for Loading Problems in a Flexible Manufacturing System". International Journal of Production Research. 1989, Vol. 27, No. 6, 1019-1034.

Sridhar, J., and Rajendran, C., "Scheduling in Cellular Manufacturing Systems: a Simulated Annealing Approach". International Journal of Production research. 1993, Vol. 31, No. 12, 2927-2945.

Srinivasan G. A, "Clustering Algorithm for Machine Cell Formation in Group Technology Using Minimum Spanning Tree". International Journal of Production Research. 1994, Vol. 32, No. 9, 2149-2158.

Srinivasan G. A, and Narendran, T. T., "GRAFICS- a Nonhierarchical Clustring Algorithm for Group Technology". International Journal of Production Research. 1991, Vol. 29, No. 3, 463-478.

Stecke, K., and Solberg, J., "Loading and Control Policies of a Flexible Manufacturing Systems". International Journal of Production Research. 1981, Vol. 19, 481-490.

Steudel, H. J. and Ballakur, A., "A Dynamic Programming Based Heuristic for

Machine Grouping in Manufacturing Cell Formation". Computers and Industrial Engineering. 1987, 12, 215-222.

Su, C.-T., and Hsu, C.-M., "Multi-Objective Machine-Part Cell Formation Through Parallel Simulated Annealing". International Journal of Production research. 1998, Vol. 36, No. 8, 2185-2207.

Sule, D. R., "Machine Capacity Planning in Group Technology". International Journal of Production Research. 1991, Vol. 29, No. 9, 1909-1922.

Sumanth, D. J., "Productivity Measurement and Evaluation Models for Manufacturing Companies". Published Doctoral Thesis, Illinois Institute of Technology, Chicago, USA. 1979.

Surer, G. A., and Ortega, M.,A., "Machine Level Based-Similarity Coefficient for Forming Manufacturing Cells". Computer and Industrial Engineering. 1994, Vol. 27, Nos. 1-4, 66-67.

Taboun S. M., Merchawi N. S., and Ulger, T., "Two-stage Model for Cost Effective Part Family and Machine Cell Formation". Computer and Industrial Engineering. 1998a, Vol. 34, No.4, 759-776.

Taboun S. M., Merchawi N. S., and Ulger, T., "Part Family and Machine Cell Formation in Multi-Period Planning Horizon of Cellular Manufacturing Systems". Production Planning & Control. 1998b, Vol. 9, No. 6, 561-571

Taboun S., M., Sankaran, S., and Bhole, S., "Comparison and Evaluation of Similarity Measures in Group Technology". Computer and Industrial Engineering. 1991, Vol. 20, No. 3, 343-353.

Taboun, S. M., Ulger, T., "Multiobjective Modeling of Operation-Allocation Problem in Flexible Manufacturing Systems". Computer and Industrial Engineering. 1992, Vol. 23, No.1-4, 295-299.

Viswanthan S., "A New Approach for Solving the P-Median Problem in GT". International Journal of Production Research . 1996, Vol. 34, No. 10, 2691-2700.

Veeramani, D., and Mani K., "A Polynomial-Time Algorithm for Optimal Clustering in a Special Class of {0,1} Matrices". International Journal of Production Research. 1996, Vol. 34, No. 9, 2587-2611.

Venugpoal, V., and Narendran T. T., "A Genetic Algorithm Approach to the Machine-Component Grouping Problem with Multiple Objectives". Computers and Industrial Engineering. 1992a, Vol. 22, No, 4, 469-480.

Venugpoal, V., and Narendran T. T., "Cell Formation in Manufacturing Systems through Simulated Annealing: an Experimental Evaluation". European Journal of Operational Research. 1992b, Vol. 63, 409-422.

Waghodeka, P. H., and Sahu, S., "Machine-Component Cell Formation in Group Technology: MACE". International Journal of Production Research. 1984, Vol. 22, No. 6, 937-948.

Wang, J., and Roze, C., "Formation of Machine Cells and Part Families: a Modified P-Median Model and a Comparative study". International Journal of Production Research. 1997, Vol. 35, No.5, 1259- 1286.

Wei, J. C., and Kern, G. M., "Commonality Analysis: a Linear Cell Clustering Algorithm for Group Technology". International Journal of Production Research. 1989, Vol. 27, No. 12, 2053-2062.

Wen, J. H., Simith, C. H., and Minor, E. D., "Fotmation and Dynamic Routing of Part Families among Flexible Manufacturing Cells". International Journal of Production Research. 1996, Vol. 34, No. 8, 2229-2245.

APPENDICES

APPENDIX 1

A SUBROTINE ADAPTED FROM MICROSOFT FORTRAN LIBRARY TO GENERATE DEMAND AND SELLING PRICE FOR ALL PROBLEMS

```
C*********************************************************************************
C*                              MAIN PROGRAM
C*********************************************************************************
      INCLUDE  'FLIB.FI'
      INCLUDE  'FLIB.FD'
      INTEGER D(50)
      INTEGER A
      OPEN (UNIT=10,STATUS='OLD',FILE='B.TXT')
      CALL SEED(RND$TIMESEED)
      DO 10 I = 1, 7
      CALL RANDOM(temp)
      A = 100*temp
      D(I)=A
      WRITE(10,100)A+200
100   FORMAT(2X,'D,I,',10I4)
10    CONTINUE
      END
```

APPENDIX 2

LINGO PROGRAM FOR THE O-1 INTEGER PROGRAMMING MODEL PRESENTED IN SECTION 3.3 USING DATA FROM PROBLEM 5 PRESENTED IN SECTION 3.5

I. INPUT FILE

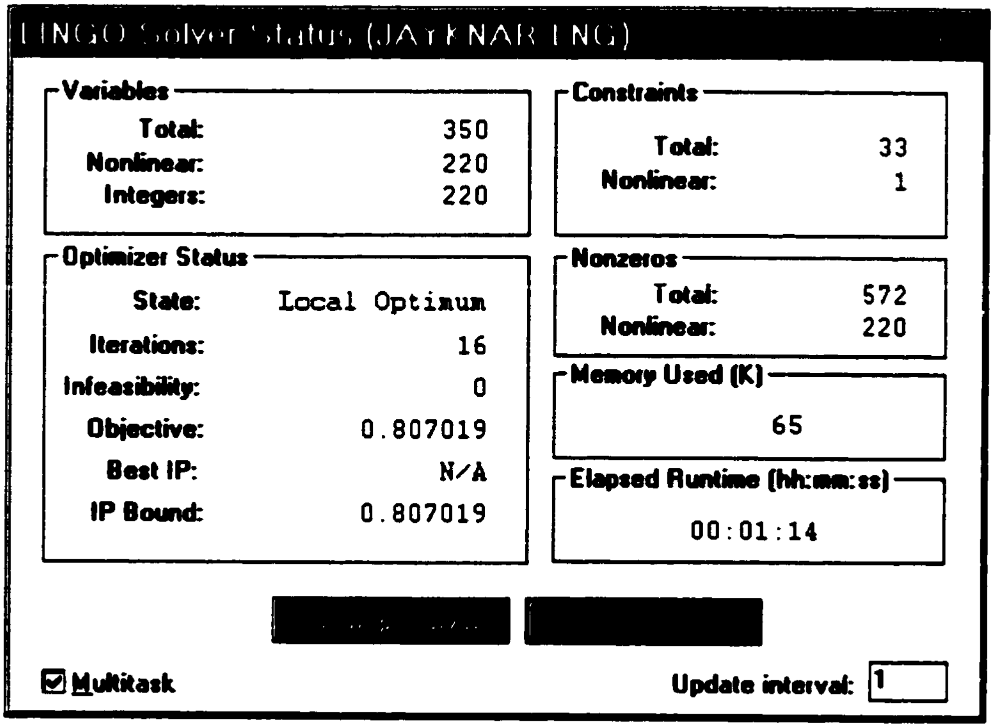
```
SETS:
!THE NUMBER OF MANUFACTURING CELLS;
CE/1..10/:;
!EACH PART "PR" HAS A NUMBER OF UNITS REQUIRED "D", NUMBER OF
MACHINES REQUIRED "NM" AND INTER-CELL HANDLING COST "EMC",
ITRA MATERIAL HANDLING COST "IMC" AND SELLING PRICE "S";
PR/1..12/:    D,IMC,EMC,S,NM;
!THE NUMBER OF MACHINES;
ME/1..10/:;
!MACHINE-PART RELATIONSHIP "PRME". E= MACHINE-PART COMPATIBILITY;
PRME(PR,ME):    E;
!PART-CELL RELATIONSHIP "PRCE" REPRESENTED BY Y;
PRCE(PR,CE):    Y;
!MACHINE-CELL RELATIONSHIP "MECE" REPRESENTED BY X. Z= NUMBER OF MACHINES
REQUIRED IN EACH CELL;
MECE(ME,CE):    X;
ENDSETS
!THE MODEL MAXIMIZES THE TOTAL PRODUCTIVITY;
MAX=
@SUM(PR(I):@SUM(CE(K):D(I)*S(I)*Y(I,K)))/
(@SUM(PR(I):@SUM(CE(K):NM(I)*IMC(I)*D(I)*Y(I,K)))+
@SUM(PR(I):@SUM(CE(K):@SUM(ME(J):D(I)*EMC(I)*(1-X(J,K))*E(I,J)*Y(I,K)))));
!THIS MODEL IS SUBJECTED TO SOME TECHNOLOGICAL CONSTRAINTS;
!MAXIMUM NUMBER OF MACHINES IN EACH CELL;
@FOR(CE(K):
@SUM(ME(J):X(J,K))<5);
!EACH MACHINE HAS TO ASSIGN TO ONLY ONE CELL;
@FOR(ME(J):
```

```
@SUM(CE(K):X(J,K))=1);
!EACH PART HAS TO ASSIGN TO ONLY ONE CELL;
@FOR(PR(I):
@SUM(CE(K):Y(I,K))=1);
!INTEGRITY;
@FOR(MECE:@BIN(X));
@FOR(PRCE:@BIN(Y));

DATA:
D=299, 291, 239, 210, 203, 281, 248, 260, 273, 299, 291, 239;
S= 14, 14, 11, 10, 10, 14, 12, 13, 13, 14, 14, 11;
IMC=2;
EMC=4.6;
NM=4,4,6,3,6,5,2,3,6,1,5,7;
E=
1, 0, 0, 1, 1, 0, 0, 0, 0, 1
,1, 0, 0, 1, 1, 0, 0, 0, 0, 1
,1, 0, 0, 1, 1, 1, 1, 0, 0, 1
, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0
, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1
, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0
, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1
, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1
, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0
, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0
, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1
, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1;

ENDDATA
END
```

II. OUTPUT FILE



ROWS= 33

VARS= 350

NO. INTEGER VARS= 220

NONLINEAR ROWS= 1

NONLINEAR VARS= 220

NONLINEAR CONSTRAINTS= 0

NONZEROS= 572

CONSTRAINT NONZ= 320

DENSITY=0.049

OPTIMAL SOLUTION FOUND AT STEP: 16

OBJECTIVE VALUE: 0.8070195

BRANCH COUNT: 0

VARIABLE	VALUE	REDUCED COST
D(1)	299.0000	0.0000000E+00

D(2)	291.0000	0.0000000E+00
D(3)	239.0000	0.0000000E+00
D(4)	210.0000	0.0000000E+00
D(5)	203.0000	0.0000000E+00
D(6)	281.0000	0.0000000E+00
D(7)	248.0000	0.0000000E+00
D(8)	260.0000	0.0000000E+00
D(9)	273.0000	0.0000000E+00
D(10)	299.0000	0.0000000E+00
D(11)	291.0000	0.0000000E+00
D(12)	239.0000	0.0000000E+00
IMC(1)	2.000000	0.0000000E+00
IMC(2)	2.000000	0.0000000E+00
IMC(3)	2.000000	0.0000000E+00
IMC(4)	2.000000	0.0000000E+00
IMC(5)	2.000000	0.0000000E+00
IMC(6)	2.000000	0.0000000E+00
IMC(7)	2.000000	0.0000000E+00
IMC(8)	2.000000	0.0000000E+00
IMC(9)	2.000000	0.0000000E+00
IMC(10)	2.000000	0.0000000E+00
IMC(11)	2.000000	0.0000000E+00
IMC(12)	2.000000	0.0000000E+00
EMC(1)	4.600000	0.0000000E+00
EMC(2)	4.600000	0.0000000E+00
EMC(3)	4.600000	0.0000000E+00
EMC(4)	4.600000	0.0000000E+00
EMC(5)	4.600000	0.0000000E+00
EMC(6)	4.600000	0.0000000E+00
EMC(7)	4.600000	0.0000000E+00
EMC(8)	4.600000	0.0000000E+00
EMC(9)	4.600000	0.0000000E+00
EMC(10)	4.600000	0.0000000E+00

EMC(11)	4.600000	0.0000000E+00
EMC(12)	4.600000	0.0000000E+00
S(1)	14.00000	0.0000000E+00
S(2)	14.00000	0.0000000E+00
S(3)	11.00000	0.0000000E+00
S(4)	10.00000	0.0000000E+00
S(5)	10.00000	0.0000000E+00
S(6)	14.00000	0.0000000E+00
S(7)	12.00000	0.0000000E+00
S(8)	13.00000	0.0000000E+00
S(9)	13.00000	0.0000000E+00
S(10)	14.00000	0.0000000E+00
S(11)	14.00000	0.0000000E+00
S(12)	11.00000	0.0000000E+00
NM(1)	4.000000	0.0000000E+00
NM(2)	4.000000	0.0000000E+00
NM(3)	6.000000	0.0000000E+00
NM(4)	3.000000	0.0000000E+00
NM(5)	6.000000	0.0000000E+00
NM(6)	5.000000	0.0000000E+00
NM(7)	2.000000	0.0000000E+00
NM(8)	3.000000	0.0000000E+00
NM(9)	6.000000	0.0000000E+00
NM(10)	1.000000	0.0000000E+00
NM(11)	5.000000	0.0000000E+00
NM(12)	7.000000	0.0000000E+00
T(1, 1)	0.0000000E+00	0.0000000E+00
T(1, 2)	0.0000000E+00	0.0000000E+00
T(1, 3)	0.0000000E+00	0.0000000E+00
T(1, 4)	0.0000000E+00	0.0000000E+00
T(1, 5)	0.0000000E+00	0.0000000E+00
T(1, 6)	0.0000000E+00	0.0000000E+00
T(1, 7)	0.0000000E+00	0.0000000E+00

T(1, 8)	0.0000000E+00	0.0000000E+00
T(1, 9)	0.0000000E+00	0.0000000E+00
T(1, 10)	0.0000000E+00	0.0000000E+00
T(2, 1)	0.0000000E+00	0.0000000E+00
T(2, 2)	0.0000000E+00	0.0000000E+00
T(2, 3)	0.0000000E+00	0.0000000E+00
T(2, 4)	0.0000000E+00	0.0000000E+00
T(2, 5)	0.0000000E+00	0.0000000E+00
T(2, 6)	0.0000000E+00	0.0000000E+00
T(2, 7)	0.0000000E+00	0.0000000E+00
T(2, 8)	0.0000000E+00	0.0000000E+00
T(2, 9)	0.0000000E+00	0.0000000E+00
T(2, 10)	0.0000000E+00	0.0000000E+00
T(3, 1)	0.0000000E+00	0.0000000E+00
T(3, 2)	0.0000000E+00	0.0000000E+00
T(3, 3)	0.0000000E+00	0.0000000E+00
T(3, 4)	0.0000000E+00	0.0000000E+00
T(3, 5)	0.0000000E+00	0.0000000E+00
T(3, 6)	0.0000000E+00	0.0000000E+00
T(3, 7)	0.0000000E+00	0.0000000E+00
T(3, 8)	0.0000000E+00	0.0000000E+00
T(3, 9)	0.0000000E+00	0.0000000E+00
T(3, 10)	0.0000000E+00	0.0000000E+00
T(4, 1)	0.0000000E+00	0.0000000E+00
T(4, 2)	0.0000000E+00	0.0000000E+00
T(4, 3)	0.0000000E+00	0.0000000E+00
T(4, 4)	0.0000000E+00	0.0000000E+00
T(4, 5)	0.0000000E+00	0.0000000E+00
T(4, 6)	0.0000000E+00	0.0000000E+00
T(4, 7)	0.0000000E+00	0.0000000E+00
T(4, 8)	0.0000000E+00	0.0000000E+00
T(4, 9)	0.0000000E+00	0.0000000E+00
T(4, 10)	0.0000000E+00	0.0000000E+00

T(5, 1)	0.0000000E+00	0.0000000E+00
T(5, 2)	0.0000000E+00	0.0000000E+00
T(5, 3)	0.0000000E+00	0.0000000E+00
T(5, 4)	0.0000000E+00	0.0000000E+00
T(5, 5)	0.0000000E+00	0.0000000E+00
T(5, 6)	0.0000000E+00	0.0000000E+00
T(5, 7)	0.0000000E+00	0.0000000E+00
T(5, 8)	0.0000000E+00	0.0000000E+00
T(5, 9)	0.0000000E+00	0.0000000E+00
T(5, 10)	0.0000000E+00	0.0000000E+00
T(6, 1)	0.0000000E+00	0.0000000E+00
T(6, 2)	0.0000000E+00	0.0000000E+00
T(6, 3)	0.0000000E+00	0.0000000E+00
T(6, 4)	0.0000000E+00	0.0000000E+00
T(6, 5)	0.0000000E+00	0.0000000E+00
T(6, 6)	0.0000000E+00	0.0000000E+00
T(6, 7)	0.0000000E+00	0.0000000E+00
T(6, 8)	0.0000000E+00	0.0000000E+00
T(6, 9)	0.0000000E+00	0.0000000E+00
T(6, 10)	0.0000000E+00	0.0000000E+00
T(7, 1)	0.0000000E+00	0.0000000E+00
T(7, 2)	0.0000000E+00	0.0000000E+00
T(7, 3)	0.0000000E+00	0.0000000E+00
T(7, 4)	0.0000000E+00	0.0000000E+00
T(7, 5)	0.0000000E+00	0.0000000E+00
T(7, 6)	0.0000000E+00	0.0000000E+00
T(7, 7)	0.0000000E+00	0.0000000E+00
T(7, 8)	0.0000000E+00	0.0000000E+00
T(7, 9)	0.0000000E+00	0.0000000E+00
T(7, 10)	0.0000000E+00	0.0000000E+00
T(8, 1)	0.0000000E+00	0.0000000E+00
T(8, 2)	0.0000000E+00	0.0000000E+00
T(8, 3)	0.0000000E+00	0.0000000E+00

T(8, 4)	0.0000000E+00	0.0000000E+00
T(8, 5)	0.0000000E+00	0.0000000E+00
T(8, 6)	0.0000000E+00	0.0000000E+00
T(8, 7)	0.0000000E+00	0.0000000E+00
T(8, 8)	0.0000000E+00	0.0000000E+00
T(8, 9)	0.0000000E+00	0.0000000E+00
T(8, 10)	0.0000000E+00	0.0000000E+00
T(9, 1)	0.0000000E+00	0.0000000E+00
T(9, 2)	0.0000000E+00	0.0000000E+00
T(9, 3)	0.0000000E+00	0.0000000E+00
T(9, 4)	0.0000000E+00	0.0000000E+00
T(9, 5)	0.0000000E+00	0.0000000E+00
T(9, 6)	0.0000000E+00	0.0000000E+00
T(9, 7)	0.0000000E+00	0.0000000E+00
T(9, 8)	0.0000000E+00	0.0000000E+00
T(9, 9)	0.0000000E+00	0.0000000E+00
T(9, 10)	0.0000000E+00	0.0000000E+00
T(10, 1)	0.0000000E+00	0.0000000E+00
T(10, 2)	0.0000000E+00	0.0000000E+00
T(10, 3)	0.0000000E+00	0.0000000E+00
T(10, 4)	0.0000000E+00	0.0000000E+00
T(10, 5)	0.0000000E+00	0.0000000E+00
T(10, 6)	0.0000000E+00	0.0000000E+00
T(10, 7)	0.0000000E+00	0.0000000E+00
T(10, 8)	0.0000000E+00	0.0000000E+00
T(10, 9)	0.0000000E+00	0.0000000E+00
T(10, 10)	0.0000000E+00	0.0000000E+00
T(11, 1)	0.0000000E+00	0.0000000E+00
T(11, 2)	0.0000000E+00	0.0000000E+00
T(11, 3)	0.0000000E+00	0.0000000E+00
T(11, 4)	0.0000000E+00	0.0000000E+00
T(11, 5)	0.0000000E+00	0.0000000E+00
T(11, 6)	0.0000000E+00	0.0000000E+00

T(11, 7)	0.0000000E+00	0.0000000E+00
T(11, 8)	0.0000000E+00	0.0000000E+00
T(11, 9)	0.0000000E+00	0.0000000E+00
T(11, 10)	0.0000000E+00	0.0000000E+00
T(12, 1)	0.0000000E+00	0.0000000E+00
T(12, 2)	0.0000000E+00	0.0000000E+00
T(12, 3)	0.0000000E+00	0.0000000E+00
T(12, 4)	0.0000000E+00	0.0000000E+00
T(12, 5)	0.0000000E+00	0.0000000E+00
T(12, 6)	0.0000000E+00	0.0000000E+00
T(12, 7)	0.0000000E+00	0.0000000E+00
T(12, 8)	0.0000000E+00	0.0000000E+00
T(12, 9)	0.0000000E+00	0.0000000E+00
T(12, 10)	0.0000000E+00	0.0000000E+00
E(1, 1)	1.000000	0.0000000E+00
E(1, 2)	0.0000000E+00	0.0000000E+00
E(1, 3)	0.0000000E+00	0.0000000E+00
E(1, 4)	1.000000	0.0000000E+00
E(1, 5)	1.000000	0.0000000E+00
E(1, 6)	0.0000000E+00	0.0000000E+00
E(1, 7)	0.0000000E+00	0.0000000E+00
E(1, 8)	0.0000000E+00	0.0000000E+00
E(1, 9)	0.0000000E+00	0.0000000E+00
E(1, 10)	1.000000	0.0000000E+00
E(2, 1)	1.000000	0.0000000E+00
E(2, 2)	0.0000000E+00	0.0000000E+00
E(2, 3)	0.0000000E+00	0.0000000E+00
E(2, 4)	1.000000	0.0000000E+00
E(2, 5)	1.000000	0.0000000E+00
E(2, 6)	0.0000000E+00	0.0000000E+00
E(2, 7)	0.0000000E+00	0.0000000E+00
E(2, 8)	0.0000000E+00	0.0000000E+00
E(2, 9)	0.0000000E+00	0.0000000E+00

E(2, 10)	1.000000	0.0000000E+00
E(3, 1)	1.000000	0.0000000E+00
E(3, 2)	0.0000000E+00	0.0000000E+00
E(3, 3)	0.0000000E+00	0.0000000E+00
E(3, 4)	1.000000	0.0000000E+00
E(3, 5)	1.000000	0.0000000E+00
E(3, 6)	1.000000	0.0000000E+00
E(3, 7)	1.000000	0.0000000E+00
E(3, 8)	0.0000000E+00	0.0000000E+00
E(3, 9)	0.0000000E+00	0.0000000E+00
E(3, 10)	1.000000	0.0000000E+00
E(4, 1)	0.0000000E+00	0.0000000E+00
E(4, 2)	0.0000000E+00	0.0000000E+00
E(4, 3)	0.0000000E+00	0.0000000E+00
E(4, 4)	0.0000000E+00	0.0000000E+00
E(4, 5)	0.0000000E+00	0.0000000E+00
E(4, 6)	1.000000	0.0000000E+00
E(4, 7)	1.000000	0.0000000E+00
E(4, 8)	0.0000000E+00	0.0000000E+00
E(4, 9)	1.000000	0.0000000E+00
E(4, 10)	0.0000000E+00	0.0000000E+00
E(5, 1)	0.0000000E+00	0.0000000E+00
E(5, 2)	1.000000	0.0000000E+00
E(5, 3)	1.000000	0.0000000E+00
E(5, 4)	0.0000000E+00	0.0000000E+00
E(5, 5)	1.000000	0.0000000E+00
E(5, 6)	1.000000	0.0000000E+00
E(5, 7)	0.0000000E+00	0.0000000E+00
E(5, 8)	1.000000	0.0000000E+00
E(5, 9)	0.0000000E+00	0.0000000E+00
E(5, 10)	1.000000	0.0000000E+00
E(6, 1)	1.000000	0.0000000E+00
E(6, 2)	0.0000000E+00	0.0000000E+00

E(6, 3)	1.000000	0.0000000E+00
E(6, 4)	0.0000000E+00	0.0000000E+00
E(6, 5)	0.0000000E+00	0.0000000E+00
E(6, 6)	1.000000	0.0000000E+00
E(6, 7)	1.000000	0.0000000E+00
E(6, 8)	1.000000	0.0000000E+00
E(6, 9)	0.0000000E+00	0.0000000E+00
E(6, 10)	0.0000000E+00	0.0000000E+00
E(7, 1)	0.0000000E+00	0.0000000E+00
E(7, 2)	0.0000000E+00	0.0000000E+00
E(7, 3)	0.0000000E+00	0.0000000E+00
E(7, 4)	0.0000000E+00	0.0000000E+00
E(7, 5)	1.000000	0.0000000E+00
E(7, 6)	0.0000000E+00	0.0000000E+00
E(7, 7)	0.0000000E+00	0.0000000E+00
E(7, 8)	0.0000000E+00	0.0000000E+00
E(7, 9)	0.0000000E+00	0.0000000E+00
E(7, 10)	1.000000	0.0000000E+00
E(8, 1)	0.0000000E+00	0.0000000E+00
E(8, 2)	1.000000	0.0000000E+00
E(8, 3)	0.0000000E+00	0.0000000E+00
E(8, 4)	0.0000000E+00	0.0000000E+00
E(8, 5)	0.0000000E+00	0.0000000E+00
E(8, 6)	1.000000	0.0000000E+00
E(8, 7)	0.0000000E+00	0.0000000E+00
E(8, 8)	0.0000000E+00	0.0000000E+00
E(8, 9)	0.0000000E+00	0.0000000E+00
E(8, 10)	1.000000	0.0000000E+00
E(9, 1)	1.000000	0.0000000E+00
E(9, 2)	1.000000	0.0000000E+00
E(9, 3)	0.0000000E+00	0.0000000E+00
E(9, 4)	0.0000000E+00	0.0000000E+00
E(9, 5)	0.0000000E+00	0.0000000E+00

E(9, 6)	1.000000	0.0000000E+00
E(9, 7)	1.000000	0.0000000E+00
E(9, 8)	1.000000	0.0000000E+00
E(9, 9)	1.000000	0.0000000E+00
E(9, 10)	0.0000000E+00	0.0000000E+00
E(10, 1)	0.0000000E+00	0.0000000E+00
E(10, 2)	0.0000000E+00	0.0000000E+00
E(10, 3)	0.0000000E+00	0.0000000E+00
E(10, 4)	0.0000000E+00	0.0000000E+00
E(10, 5)	0.0000000E+00	0.0000000E+00
E(10, 6)	0.0000000E+00	0.0000000E+00
E(10, 7)	0.0000000E+00	0.0000000E+00
E(10, 8)	1.000000	0.0000000E+00
E(10, 9)	0.0000000E+00	0.0000000E+00
E(10, 10)	0.0000000E+00	0.0000000E+00
E(11, 1)	1.000000	0.0000000E+00
E(11, 2)	1.000000	0.0000000E+00
E(11, 3)	0.0000000E+00	0.0000000E+00
E(11, 4)	1.000000	0.0000000E+00
E(11, 5)	0.0000000E+00	0.0000000E+00
E(11, 6)	0.0000000E+00	0.0000000E+00
E(11, 7)	1.000000	0.0000000E+00
E(11, 8)	0.0000000E+00	0.0000000E+00
E(11, 9)	0.0000000E+00	0.0000000E+00
E(11, 10)	1.000000	0.0000000E+00
E(12, 1)	1.000000	0.0000000E+00
E(12, 2)	1.000000	0.0000000E+00
E(12, 3)	1.000000	0.0000000E+00
E(12, 4)	0.0000000E+00	0.0000000E+00
E(12, 5)	1.000000	0.0000000E+00
E(12, 6)	1.000000	0.0000000E+00
E(12, 7)	0.0000000E+00	0.0000000E+00
E(12, 8)	1.000000	0.0000000E+00

E(12, 9)	0.0000000E+00	0.0000000E+00
E(12, 10)	1.000000	0.0000000E+00
Y(1, 1)	0.0000000E+00	0.0000000E+00
Y(1, 2)	0.0000000E+00	0.0000000E+00
Y(1, 3)	0.0000000E+00	0.0000000E+00
Y(1, 4)	0.0000000E+00	0.0000000E+00
Y(1, 5)	0.0000000E+00	0.0000000E+00
Y(1, 6)	1.000000	0.0000000E+00
Y(1, 7)	0.0000000E+00	0.0000000E+00
Y(1, 8)	0.0000000E+00	0.0000000E+00
Y(1, 9)	0.0000000E+00	0.0000000E+00
Y(1, 10)	0.0000000E+00	0.0000000E+00
Y(2, 1)	0.0000000E+00	0.0000000E+00
Y(2, 2)	0.0000000E+00	0.0000000E+00
Y(2, 3)	0.0000000E+00	0.0000000E+00
Y(2, 4)	0.0000000E+00	0.0000000E+00
Y(2, 5)	0.0000000E+00	0.0000000E+00
Y(2, 6)	1.000000	0.0000000E+00
Y(2, 7)	0.0000000E+00	0.0000000E+00
Y(2, 8)	0.0000000E+00	0.0000000E+00
Y(2, 9)	0.0000000E+00	0.0000000E+00
Y(2, 10)	0.0000000E+00	0.0000000E+00
Y(3, 1)	0.0000000E+00	0.0000000E+00
Y(3, 2)	0.0000000E+00	0.0000000E+00
Y(3, 3)	0.0000000E+00	0.0000000E+00
Y(3, 4)	0.0000000E+00	0.0000000E+00
Y(3, 5)	0.0000000E+00	0.0000000E+00
Y(3, 6)	1.000000	0.0000000E+00
Y(3, 7)	0.0000000E+00	0.0000000E+00
Y(3, 8)	0.0000000E+00	0.0000000E+00
Y(3, 9)	0.0000000E+00	0.0000000E+00
Y(3, 10)	0.0000000E+00	0.0000000E+00
Y(4, 1)	0.0000000E+00	0.0000000E+00

Y(4, 2)	0.0000000E+00	0.0000000E+00
Y(4, 3)	0.0000000E+00	0.0000000E+00
Y(4, 4)	0.0000000E+00	0.0000000E+00
Y(4, 5)	0.0000000E+00	0.0000000E+00
Y(4, 6)	1.000000	0.0000000E+00
Y(4, 7)	0.0000000E+00	0.0000000E+00
Y(4, 8)	0.0000000E+00	0.0000000E+00
Y(4, 9)	0.0000000E+00	0.0000000E+00
Y(4, 10)	0.0000000E+00	0.0000000E+00
Y(5, 1)	0.0000000E+00	0.0000000E+00
Y(5, 2)	0.0000000E+00	0.0000000E+00
Y(5, 3)	0.0000000E+00	0.0000000E+00
Y(5, 4)	0.0000000E+00	0.0000000E+00
Y(5, 5)	0.0000000E+00	0.0000000E+00
Y(5, 6)	1.000000	0.0000000E+00
Y(5, 7)	0.0000000E+00	0.0000000E+00
Y(5, 8)	0.0000000E+00	0.0000000E+00
Y(5, 9)	0.0000000E+00	0.0000000E+00
Y(5, 10)	0.0000000E+00	0.0000000E+00
Y(6, 1)	0.0000000E+00	0.0000000E+00
Y(6, 2)	0.0000000E+00	0.0000000E+00
Y(6, 3)	0.0000000E+00	0.0000000E+00
Y(6, 4)	0.0000000E+00	0.0000000E+00
Y(6, 5)	0.0000000E+00	0.0000000E+00
Y(6, 6)	1.000000	0.0000000E+00
Y(6, 7)	0.0000000E+00	0.0000000E+00
Y(6, 8)	0.0000000E+00	0.0000000E+00
Y(6, 9)	0.0000000E+00	0.0000000E+00
Y(6, 10)	0.0000000E+00	0.0000000E+00
Y(7, 1)	0.0000000E+00	0.0000000E+00
Y(7, 2)	0.0000000E+00	0.0000000E+00
Y(7, 3)	0.0000000E+00	0.0000000E+00
Y(7, 4)	0.0000000E+00	0.0000000E+00

Y(7, 5)	0.0000000E+00	0.0000000E+00
Y(7, 6)	1.000000	0.0000000E+00
Y(7, 7)	0.0000000E+00	0.0000000E+00
Y(7, 8)	0.0000000E+00	0.0000000E+00
Y(7, 9)	0.0000000E+00	0.0000000E+00
Y(7, 10)	0.0000000E+00	0.0000000E+00
Y(8, 1)	0.0000000E+00	0.0000000E+00
Y(8, 2)	0.0000000E+00	0.0000000E+00
Y(8, 3)	0.0000000E+00	0.0000000E+00
Y(8, 4)	0.0000000E+00	0.0000000E+00
Y(8, 5)	0.0000000E+00	0.0000000E+00
Y(8, 6)	1.000000	0.0000000E+00
Y(8, 7)	0.0000000E+00	0.0000000E+00
Y(8, 8)	0.0000000E+00	0.0000000E+00
Y(8, 9)	0.0000000E+00	0.0000000E+00
Y(8, 10)	0.0000000E+00	0.0000000E+00
Y(9, 1)	0.0000000E+00	0.0000000E+00
Y(9, 2)	0.0000000E+00	0.0000000E+00
Y(9, 3)	0.0000000E+00	0.0000000E+00
Y(9, 4)	0.0000000E+00	0.0000000E+00
Y(9, 5)	0.0000000E+00	0.0000000E+00
Y(9, 6)	1.000000	0.0000000E+00
Y(9, 7)	0.0000000E+00	0.0000000E+00
Y(9, 8)	0.0000000E+00	0.0000000E+00
Y(9, 9)	0.0000000E+00	0.0000000E+00
Y(9, 10)	0.0000000E+00	0.0000000E+00
Y(10, 1)	0.0000000E+00	0.0000000E+00
Y(10, 2)	0.0000000E+00	0.0000000E+00
Y(10, 3)	0.0000000E+00	0.0000000E+00
Y(10, 4)	0.0000000E+00	0.0000000E+00
Y(10, 5)	0.0000000E+00	0.0000000E+00
Y(10, 6)	1.000000	0.0000000E+00
Y(10, 7)	0.0000000E+00	0.0000000E+00

Y(10, 8)	0.0000000E+00	0.0000000E+00
Y(10, 9)	0.0000000E+00	0.0000000E+00
Y(10, 10)	0.0000000E+00	0.0000000E+00
Y(11, 1)	0.0000000E+00	0.0000000E+00
Y(11, 2)	0.0000000E+00	0.0000000E+00
Y(11, 3)	0.0000000E+00	0.0000000E+00
Y(11, 4)	0.0000000E+00	0.0000000E+00
Y(11, 5)	0.0000000E+00	0.0000000E+00
Y(11, 6)	1.000000	0.0000000E+00
Y(11, 7)	0.0000000E+00	0.0000000E+00
Y(11, 8)	0.0000000E+00	0.0000000E+00
Y(11, 9)	0.0000000E+00	0.0000000E+00
Y(11, 10)	0.0000000E+00	0.0000000E+00
Y(12, 1)	0.0000000E+00	0.0000000E+00
Y(12, 2)	0.0000000E+00	0.0000000E+00
Y(12, 3)	0.0000000E+00	0.0000000E+00
Y(12, 4)	0.0000000E+00	0.0000000E+00
Y(12, 5)	0.0000000E+00	0.0000000E+00
Y(12, 6)	1.000000	0.0000000E+00
Y(12, 7)	0.0000000E+00	0.0000000E+00
Y(12, 8)	0.0000000E+00	0.0000000E+00
Y(12, 9)	0.0000000E+00	0.0000000E+00
Y(12, 10)	0.0000000E+00	0.0000000E+00
X(1, 1)	0.0000000E+00	0.5404316E-01
X(1, 2)	0.0000000E+00	0.0000000E+00
X(1, 3)	0.0000000E+00	0.0000000E+00
X(1, 4)	0.0000000E+00	0.0000000E+00
X(1, 5)	0.0000000E+00	0.0000000E+00
X(1, 6)	1.000000	0.0000000E+00
X(1, 7)	0.0000000E+00	0.0000000E+00
X(1, 8)	0.0000000E+00	0.0000000E+00
X(1, 9)	0.0000000E+00	0.0000000E+00
X(1, 10)	0.0000000E+00	0.0000000E+00

X(2, 1)	1.000000	0.0000000E+00
X(2, 2)	0.0000000E+00	0.0000000E+00
X(2, 3)	0.0000000E+00	0.0000000E+00
X(2, 4)	0.0000000E+00	0.0000000E+00
X(2, 5)	0.0000000E+00	0.0000000E+00
X(2, 6)	0.0000000E+00	0.0000000E+00
X(2, 7)	0.0000000E+00	0.0000000E+00
X(2, 8)	0.0000000E+00	0.0000000E+00
X(2, 9)	0.0000000E+00	0.0000000E+00
X(2, 10)	0.0000000E+00	0.0000000E+00
X(3, 1)	1.000000	0.0000000E+00
X(3, 2)	0.0000000E+00	0.0000000E+00
X(3, 3)	0.0000000E+00	0.0000000E+00
X(3, 4)	0.0000000E+00	0.0000000E+00
X(3, 5)	0.0000000E+00	0.0000000E+00
X(3, 6)	0.0000000E+00	0.0000000E+00
X(3, 7)	0.0000000E+00	0.0000000E+00
X(3, 8)	0.0000000E+00	0.0000000E+00
X(3, 9)	0.0000000E+00	0.0000000E+00
X(3, 10)	0.0000000E+00	0.0000000E+00
X(4, 1)	1.000000	0.0000000E+00
X(4, 2)	0.0000000E+00	0.0000000E+00
X(4, 3)	0.0000000E+00	0.0000000E+00
X(4, 4)	0.0000000E+00	0.0000000E+00
X(4, 5)	0.0000000E+00	0.0000000E+00
X(4, 6)	0.0000000E+00	0.0000000E+00
X(4, 7)	0.0000000E+00	0.0000000E+00
X(4, 8)	0.0000000E+00	0.0000000E+00
X(4, 9)	0.0000000E+00	0.0000000E+00
X(4, 10)	0.0000000E+00	0.0000000E+00
X(5, 1)	0.0000000E+00	0.0000000E+00
X(5, 2)	0.0000000E+00	0.0000000E+00
X(5, 3)	0.0000000E+00	0.0000000E+00

X(5, 4)	0.0000000E+00	0.0000000E+00
X(5, 5)	0.0000000E+00	0.0000000E+00
X(5, 6)	1.000000	0.0000000E+00
X(5, 7)	0.0000000E+00	0.0000000E+00
X(5, 8)	0.0000000E+00	0.0000000E+00
X(5, 9)	0.0000000E+00	0.0000000E+00
X(5, 10)	0.0000000E+00	0.0000000E+00
X(6, 1)	0.0000000E+00	0.0000000E+00
X(6, 2)	0.0000000E+00	0.0000000E+00
X(6, 3)	0.0000000E+00	0.0000000E+00
X(6, 4)	0.0000000E+00	0.0000000E+00
X(6, 5)	0.0000000E+00	0.0000000E+00
X(6, 6)	1.000000	0.0000000E+00
X(6, 7)	0.0000000E+00	0.0000000E+00
X(6, 8)	0.0000000E+00	0.0000000E+00
X(6, 9)	0.0000000E+00	0.0000000E+00
X(6, 10)	0.0000000E+00	0.0000000E+00
X(7, 1)	0.0000000E+00	0.0000000E+00
X(7, 2)	0.0000000E+00	0.0000000E+00
X(7, 3)	0.0000000E+00	0.0000000E+00
X(7, 4)	0.0000000E+00	0.0000000E+00
X(7, 5)	0.0000000E+00	0.0000000E+00
X(7, 6)	0.0000000E+00	0.0000000E+00
X(7, 7)	0.0000000E+00	0.0000000E+00
X(7, 8)	0.0000000E+00	0.0000000E+00
X(7, 9)	0.0000000E+00	0.0000000E+00
X(7, 10)	1.000000	0.0000000E+00
X(8, 1)	0.0000000E+00	0.0000000E+00
X(8, 2)	0.0000000E+00	0.0000000E+00
X(8, 3)	0.0000000E+00	0.0000000E+00
X(8, 4)	0.0000000E+00	0.0000000E+00
X(8, 5)	0.0000000E+00	0.0000000E+00
X(8, 6)	1.000000	0.0000000E+00

X(8, 7)	0.0000000E+00	0.0000000E+00
X(8, 8)	0.0000000E+00	0.0000000E+00
X(8, 9)	0.0000000E+00	0.0000000E+00
X(8, 10)	0.0000000E+00	0.0000000E+00
X(9, 1)	0.0000000E+00	0.0000000E+00
X(9, 2)	1.000000	0.0000000E+00
X(9, 3)	0.0000000E+00	0.0000000E+00
X(9, 4)	0.0000000E+00	0.0000000E+00
X(9, 5)	0.0000000E+00	0.0000000E+00
X(9, 6)	0.0000000E+00	0.0000000E+00
X(9, 7)	0.0000000E+00	0.0000000E+00
X(9, 8)	0.0000000E+00	0.0000000E+00
X(9, 9)	0.0000000E+00	0.0000000E+00
X(9, 10)	0.0000000E+00	0.0000000E+00
X(10, 1)	0.0000000E+00	0.0000000E+00
X(10, 2)	0.0000000E+00	0.0000000E+00
X(10, 3)	0.0000000E+00	0.0000000E+00
X(10, 4)	0.0000000E+00	0.0000000E+00
X(10, 5)	0.0000000E+00	0.0000000E+00
X(10, 6)	1.000000	0.0000000E+00
X(10, 7)	0.0000000E+00	0.0000000E+00
X(10, 8)	0.0000000E+00	0.0000000E+00
X(10, 9)	0.0000000E+00	0.0000000E+00
X(10, 10)	0.0000000E+00	0.0000000E+00

ROW	SLACK OR SURPLUS	DUAL PRICE
1	0.8070195	0.0000000E+00
2	2.000000	0.0000000E+00
3	4.000000	0.0000000E+00
4	5.000000	0.0000000E+00
5	5.000000	0.0000000E+00
6	5.000000	0.0000000E+00
7	0.0000000E+00	0.0000000E+00

8	5.000000	0.0000000E+00
9	5.000000	0.0000000E+00
10	5.000000	0.0000000E+00
11	4.000000	0.0000000E+00
12	0.0000000E+00	0.0000000E+00
13	0.0000000E+00	0.0000000E+00
14	0.0000000E+00	0.0000000E+00
15	0.0000000E+00	0.0000000E+00
16	0.0000000E+00	0.0000000E+00
17	0.0000000E+00	0.0000000E+00
18	0.0000000E+00	0.0000000E+00
19	0.0000000E+00	0.0000000E+00
20	0.0000000E+00	0.0000000E+00
21	0.0000000E+00	0.0000000E+00
22	0.0000000E+00	0.0000000E+00
23	0.0000000E+00	0.0000000E+00
24	0.0000000E+00	0.0000000E+00
25	0.0000000E+00	0.0000000E+00
26	0.0000000E+00	0.0000000E+00
27	0.0000000E+00	0.0000000E+00
28	0.0000000E+00	0.0000000E+00
29	0.0000000E+00	0.0000000E+00
30	0.0000000E+00	0.0000000E+00
31	0.0000000E+00	0.0000000E+00
32	0.0000000E+00	0.0000000E+00
33	0.0000000E+00	0.0000000E+00

APPENDIX 3

COMPUTER PROGRAM FOR THE SIMULATED ANNEALING ALGORITHM PRESENTED IN SECTION 4.2 USING DATA FROM PROBLEM 1 PRESENTED IN SECTION 4.3

```
C*****************************************************************************************
C*                    LIST OF THE VARIABLES USED IN THIS PROGRAM
C*****************************************************************************************
C    BTP        BEST TOTAL PRODUCTIVITY FOUND SO FAR
C    BX         BEST SOLUTION FOUND
C    C          NUMBER OF CELLS
C    D          DEMAND (NUMBER OF UNITS REQUIRED)
C    EMC        INTER CELL COST
C    IMC        INTRA CELL COST
C    M          THE NUMBER OF MACHINES IN THE PROBLEM
C    MMAX       MAXIMUM NUMBER OF MACHINES IN EACH CELL
C    NIRAT      NUMBER OF ITERATION AT EACH TEMPERATURE LEVEL
C    NTP        NEIGHBOR SOLUTION (NEW TOTAL PRODUCTIVITY)
C    R1         THE SELECTED MACHINE TO MOVE (RANDOMLY)
C    R2         THE SELECTED CELL  (RANDOMLY)
C    O          OUTPUT
C    P          NUMBER OF PARTS IN THE PROBLEM
C    S          SELLING PRICE
C    SUMI       TOTAL NUMBER OF INTER CELL MOVEMENT
C    SUME       TOTAL NUMBER OF INTRA CELL MOVEMENT
C    SUMT       TOTAL MATERIAL HANDLING COST
C    T          CURRENT TEMPERATURE  (CURRENT SOLUTION)
C    TP         CURRENT TOTAL PRODUCTIVITY
C    TSTOP      STOPPING TEMPERATURE
C    TFAC       DECREMANTAL FACTOR
```

```
C*****************************************************************************************
C*                              MIN PROGRAM                                              *
C*****************************************************************************************
INTEGER Y(150,150),X(150,150),D(150),S(150),NX(150,150),MC(150),
    SUMI(130),SUME(130), SUM1,SUM2,SUMT,SUM(130),PR(130),LSUM(130),
    BX(130,130)
    REAL NTP,T
    N1=586754699
    N2=326574579
    N3=865473439
C  INPUTS
    MAXM=4
    EMC=4.6
    IMC=2
    T=0.5
    NIRAT=7
    TFAC=0.98
    TSTOP=0.05

C*****************************************************************************************
C*                     OPEN, READ AND WRITE INPUT FILE                                   *
C*****************************************************************************************
C   OPEN THE INPUT FIEL WHICH INCLUDE THE FOLLOWING
C   NUMBER OF PARTS, MACHINES AND CELLS
C   PART-MACHINE RELATION SHIP
C   MACHINE-CELL RELATIONSHIP"
C    DEMAND
C   SELLING PRICE
     OPEN (UNIT=20,FILE='P12N.FOR')
     OPEN (UNIT=10,STATUS='OLD',FILE='ABR.TXT')
     READ(20,*)P,M,C
     DO 10 I=1,P
        READ(20,*)(Y(I,J),J=1,M)
```

```
 10 CONTINUE
      DO 20 I=1,P
         WRITE (*,*)(Y(I,J),J=1,M)
 20  CONTINUE
      DO 30 K=1,C
         READ (20,*)(X(K,J),J=1,M)
 30  CONTINUE
      DO 40 K=1,C
         WRITE (10,*)(X(K,J),J=1,M)
 40  CONTINUE
      READ (20,*)(D(I),I=1,P)
      WRITE(10,*)(D(I),I=1,P)
      READ (20,*)(S(I),I=1,P)
      WRITE (10,*)(S(I),I=1,P)

C*****************************************************************************
C*                    CALCULATE THE INITIAL SOLUTION                          *
C*****************************************************************************
      OUT=0
      DO 50 I=1,P
         OUT=OUT+D(I)*S(I)
         WRITE(10,*)'OUT=',OUT
 50  CONTINUE
      WRITE(*,*)'OUTPUT=',OUT
      SUMT=0
      DO 60  I=1,P
         SUMI(I)=0
         SUME(I)=0
         SUM1=0
         SUM2=0
         DO 70  K=1,C
            NSW=0
            DO 80 J=1,M
```

```
                 IF((Y(I,J).EQ.1).AND.(X(K,J).EQ.1))THEN
                 SUM1=SUM1+1
                 NSW=1
                 END IF
80            CONTINUE
              IF(NSW.EQ.1)THEN
              SUM2=SUM2+1
              END IF
70         CONTINUE
           SUMI(I)=SUM1
           SUME(I)=SUM2-1
           WRITE(10,*)'SUMI(',I,')=',SUMI(I)
           WRITE(10,*)'SUME(',I,')=',SUME(I)
           SUMT=SUMT+(SUMI(I))*D(I)*IMC+SUME(I)*D(I)*EMC
           WRITE(10,*)'SUMT=',SUMT
60      CONTINUE
        TP=OUT/SUMT
        WRITE(10,*)'TP=',TP
        BTP=TP

C*************************************************************************************
C*                          GENERATE A NEW SOLUTION                                  *
C*************************************************************************************
        DO 90 K=1,C
           DO 100 J=1,M
              IF (X(K,J).EQ.1) MC(J)=K
                  NX(K,J)=X(K,J)
                  BX(K,J)=X(K,J)
100     CONTINUE
90      CONTINUE
205     IR=0
120     IR=IR+1
```

```
C*******************************************************************************
C*          THE LINES FROM STATEMENT LABEL 130 TO LABEL 150 ARE ADAPTED        *
C*                              FROM COTRUVO(1993)                             *
C*******************************************************************************
130 CALL RAND1(N1,RD1)
    R1 =INT(M*RD1 + 1)
    WRITE(10,*)'THE MACHINE TO MOVE =',R1
    CALL RAND2(N2,RD2)
    R2 =INT(C*RD2 + 1)
    WRITE(10,*)'THE CELL TO MOVE THE MACHINE TO  =',R2
    IF (MC(R1).EQ.R2) GOTO 130
    SE=0
    DO 140 J=1,M
       SE = SE + NX(R2,J)
140 CONTINUE
    IF (SE.GE.MAXM) GOTO 130
    NX(MC(R1),R1) =0
    NX(R2,R1) = 1
    MC(R1)=R2
    DO 150 K=1,C
       WRITE(10,*)(NX(K,J),J=1,M)
150 CONTINUE
    DO 160 I=1,P
       SUMI(I)=0
       SUME(I)=0
       SUM1=0
       SUM2=0
       DO 170  K=1,C
               NSW=0
               DO 180 J=1,M
       IF((Y(I,J).EQ.1).AND.(NX(K,J).EQ.1))THEN
       SUM1=SUM1+1
       NSW=1
```

```
          END IF
 180 CONTINUE
       IF(NSW.EQ.1)THEN
       SUM2=SUM2+1
       END IF
 170 CONTINUE
       SUMI(I)=SUM1
       SUME(I)=SUM2-1
       WRITE(10,*)'SUMI(',I,')=',SUMI(I)
       WRITE(10,*)'SUME(',I,')=',SUME(I)
       SUMT=SUMT+(SUMI(I))*D(I)*IMC+SUME(I)*D(I)*EMC
       WRITE(10,*)'SUMT=',SUMT
 160 CONTINUE
       NTP=OUT/SUMT
       WRITE(10,*)'NTP=',NTP
       IF(NTP.GE.TP)THEN
       TP=NTP
       NX(MC(R1),R1)=0
       NX(R2,R1)=1
       MC(R1)=R2
       ENDIF
       IF(NTP.GE.BTP)THEN
          BTP=NTP
          NX(MC(R1),R1)=0
          NX(R2,R1)=1
          MC(R1)=R2
          DO 190 K=1,C
             DO 200 J=1,M
                BX(K,J)=NX(K,J)
200 CONTINUE
190 CONTINUE
      ENDIF
      IF(NTP.LT.TP)THEN
```

```
      DZ=ABS(NTP-TP)
      ACC=EXP(-DZ/T)
      WRITE(10,*)'ACC=',ACC
      CALL RAND3(N3,RD3)
      WRITE(10,*)'RD3=',RD3
      IF(RD3.LE.ACC)THEN
           TP=NTP
           NX(MC(R1),R1)=0
           NX(R2,R1)=1
           MC(R1)=R2
      ELSE
           NTP=TP
           NX(MC(R1),R1)=0
           NX(R2,R1)=1
      ENDIF
    ENDIF
    IF(IR.LT.MNIRAT)GOTO 120
195 T=T*TFAC
    IF(T.LE.TSTOP)THEN
      GOTO 200
    ELSE
      GOTO 205
    ENDIF
200 DO 210 K=1,C
      WRITE(10,*)(BX(K,J),J=1,M)
210 CONTINUE
    WRITE (10,*)'PRODUCTIVITY=',BTP
    WRITE (10,*)
    WRITE (10,215)
    DO 220 K=1,C
      WRITE (10,225)K
      DO 230 J=1,M
           IF(BX(K,J).EQ.1)WRITE(10,235)J
```

```
230    CONTINUE
220 CONTINUE
215 FORMAT(2X,'CELL',4X,'MACHINE')
225 FORMAT(2X,I2)
235 FORMAT (10X,I2)

C*****************************************************************************************
C*                                  PART ASSINMENT
C*****************************************************************************************
      DO 240 I=1,P
         DO 250 K=1,C
                SUM(K)=0
                SUM1=0
                DO 280 J=1,M
         IF ((Y(I,J).EQ.1).AND.(BX(K,J).EQ.1)) THEN
         SUM1=SUM1+1
         END IF
260             CONTINUE
      SUM(K)=SUM1
250    CONTINUE
      LSUM(I)=SUM(1)
      PR(I)=1
      DO 270 K=2,C
         IF (SUM(K).GT.LSUM(I))THEN
         LSUM(I)=SUM(K)
         PR(I)=K
         END IF
270 CONTINUE
240  CONTINUE
      WRITE (10,275)
      DO 280 K=1,C
         WRITE(10,285)K
         DO 300 I=1,P
```

```
               IF(PR(I).EQ.K)WRITE(10,305)I
300      CONTINUE
280 CONTINUE
275 FORMAT(2X,'CELL',4X,'PART')
285 FORMAT(1X,I2)
305 FORMAT(8X,I3)
      END

C*****************************************************************************************
C*            THE THE FOLLOWING SUBROUTINES ARE ADAPTED FROM ARE ADAPTED     *
C*                              FROM COTRUVO(1993)                               *
C*****************************************************************************************
      SUBROUTINE RAND1(N1,RD1)
      L=316227
      N1=N1*L
      RN=N1
      RD1=RN/(2.**31 - 1)
      RD1=ABS(RD1)
      RETURN
      END
      SUBROUTINE RAND2(N2,RD2)
      L=316227
      N2=N2*L
      RN=N2
      RD2=RN/(2.**31 - 1)
      RD2=ABS(RD2)
      RETURN
      END
      SUBROUTINE RAND3(N3,RD3)
      L=316227
      N3 =N3*L
      RN=N3
      RD3 =RN/(2.**31 - 1)
```

```
RD3 = ABS(RD3)
RETURN
END
```

APPENDIX 4

LINGO PROGRAM FOR THE O-1 INTEGER PROGRAMMING MODEL PRESENTED IN SECTION 5.3 USING DATA FROM PROBLEM 1 PRESENTED IN SECTION 5.2

```
SETS:
!THE NUMBER OF MANUFACTURING CELLS;
CE/1..4/:;
!EACH PART "PR" HAS A OPTIMUM DEMAND REQUIRED "D", NUMBER OF
MACHINES REQUIRED "NM" AND INTER-CELL HANDLING COST "EMC",
ITRA MATERIAL HANDLING COST "IMC" AND SELLING PRICE "S";
PR/1..5/:    D,IMC,EMC,S;
!THE NUMBER OF PROCESS PLANS;
PP/1..3/:;
!THE NUMBER OF MACHINES;
ME/1..4/:    B;
!MACHINE-PAERT AND PROCESS PLAN RELATIONSHIP "PRME". E= MACHINE-
PARTAND PROCESS PLNE COMPATIBILITY;
PRPPME(PR,PP,ME):    E;
!PART-CELL AND PROCESS PLNE RELATIONSHIP"PRPP.  NM IS THE NUMBER OF
MACHINES REQUIRED BY PART I;
PRPP(PR,PP):NM;
!PART-PROCESS PLAN AND CELL RELATIONSHIP "PRCE" REPRESENTED BY Y;
PRPPCE(PR,PP,CE):    Y;
!MACHINE-CELL RELATIONSHIP "PRCE" REPRESENTED BY X;
MECE(ME,CE):    X;
ENDSETS
!THE MODEL MAXIMIZES THE TOTAL PRODUCTIVITY;
MAX= @SUM(PR(I):@SUM(PP(R):@SUM(CE(K):D(I)*S(I)*Y(I,R,K))))/
(@SUM(PR(I):@SUM(PP(R):  @SUM(CE(K):NM(I,R)*IMC(I)*D(I)*Y(I,R,K))))+
@SUM(PR(I):@SUM(PP(R):@SUM(ME(J):@SUM(CE(K):D(I)*EMC(I)*(1-X(J,K))
*E(I,R,J)*Y(I,R,K))))));
!THIS MODEL IS SUBJECTED TO SOME TECHNOLOGICAL CONSTRAINTS;
!MAXIMUM NUMBER OF MACHINES IN EACH CELL;
```

```
@FOR(CE(K): @SUM(ME(J):X(J,K))<2);
!EACH MACHINE HAS TO ASSIGN TO ONLY ONE CELL;
@FOR(ME(J):
        @SUM(CE(K):X(J,K))=1);
!EACH PART HAS TO ASSIGN TO ONLY ONE CELL;
@FOR(PR(I):
        @SUM(CE(K):@SUM(PP(R):Y(I,R,K)))=1);
!INTEGRITY;
@FOR(MECE:@BIN(X));
@FOR(PRPPCE:@BIN(Y));
DATA:
D= 239, 210, 203, 248, 260;
S= 10, 10, 14, 12, 13;
IMC=2;
EMC=4.6;
NM=  2,2,2,
     2,2,10,
     2,2,10,
     2,2,10,
     2,1,10;
E=
     0,0,1,1,
     0,1,0,1,
     1,1,0,0,
     0,1,1,0,
     1,0,1,0,
     0,0,0,0,
     1,0,0,1,
     0,1,0,1,
     0,0,0,0,
     1,0,0,1,
     1,0,1,0,
     0,0,0,0,
```

0,0,1,1,

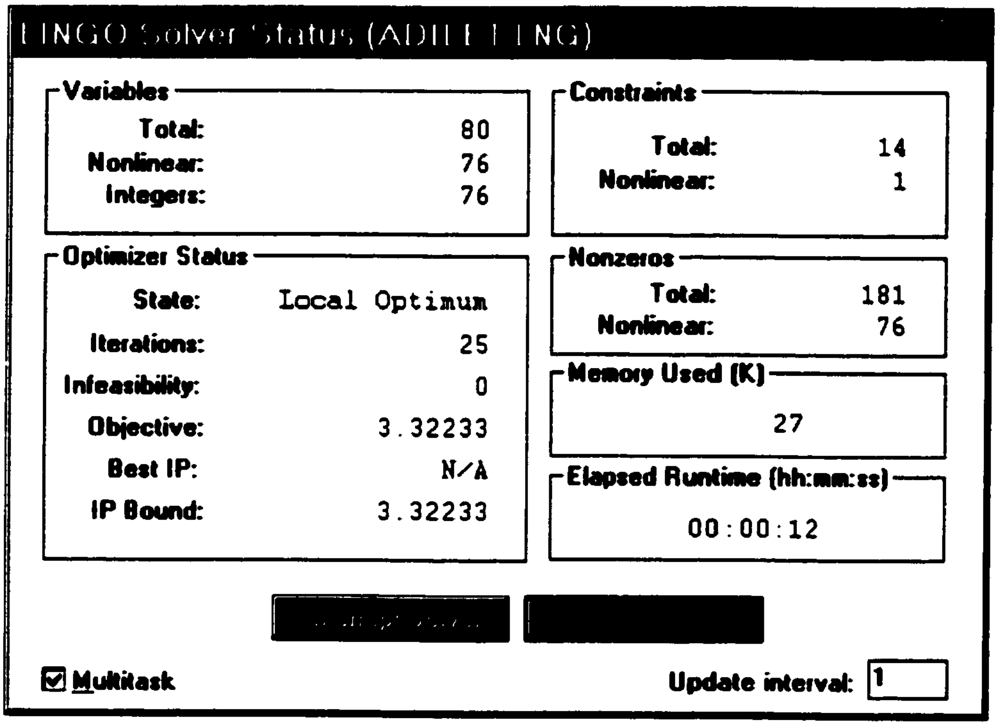
1,0,0,0,

0,0,0,0;

ENDDATA

END

II OUTPUT FILE



ROWS= 14

VARS= 80

NO. INTEGER VARS= 76

NONLINEAR ROWS= 1

NONLINEAR VARS= 76

NONLINEAR CONSTRAINTS= 0

NONZEROS= 181

CONSTRAINT NONZ= 92

DENSITY=0.160

OPTIMAL SOLUTION FOUND AT STEP: 25

TPECTIVE VALUE: 3.322330

BRANCH COUNT: 0

VARIABLE	VALUE	REDUCED COST
D(1)	239.0000	0.0000000E+00
D(2)	210.0000	0.0000000E+00
D(3)	203.0000	0.0000000E+00
D(4)	248.0000	0.0000000E+00
D(5)	260.0000	0.0000000E+00
IMC(1)	2.000000	0.0000000E+00
IMC(2)	2.000000	0.0000000E+00
IMC(3)	2.000000	0.0000000E+00
IMC(4)	2.000000	0.0000000E+00
IMC(5)	2.000000	0.0000000E+00
EMC(1)	4.600000	0.0000000E+00
EMC(2)	4.600000	0.0000000E+00
EMC(3)	4.600000	0.0000000E+00
EMC(4)	4.600000	0.0000000E+00
EMC(5)	4.600000	0.0000000E+00
S(1)	10.00000	0.0000000E+00
S(2)	10.00000	0.0000000E+00
S(3)	14.00000	0.0000000E+00
S(4)	12.00000	0.0000000E+00
S(5)	13.00000	0.0000000E+00
B(1)	0.0000000E+00	0.0000000E+00
B(2)	0.0000000E+00	0.0000000E+00
B(3)	0.0000000E+00	0.0000000E+00
B(4)	0.0000000E+00	0.0000000E+00
E(1, 1, 1)	0.0000000E+00	0.0000000E+00
E(1, 1, 2)	0.0000000E+00	0.0000000E+00
E(1, 1, 3)	1.000000	0.0000000E+00
E(1, 1, 4)	1.000000	0.0000000E+00
E(1, 2, 1)	0.0000000E+00	0.0000000E+00

E(1, 2, 2)	1.000000	0.0000000E+00
E(1, 2, 3)	0.0000000E+00	0.0000000E+00
E(1, 2, 4)	1.000000	0.0000000E+00
E(1, 3, 1)	1.000000	0.0000000E+00
E(1, 3, 2)	1.000000	0.0000000E+00
E(1, 3, 3)	0.0000000E+00	0.0000000E+00
E(1, 3, 4)	0.0000000E+00	0.0000000E+00
E(2, 1, 1)	0.0000000E+00	0.0000000E+00
E(2, 1, 2)	1.000000	0.0000000E+00
E(2, 1, 3)	1.000000	0.0000000E+00
E(2, 1, 4)	0.0000000E+00	0.0000000E+00
E(2, 2, 1)	1.000000	0.0000000E+00
E(2, 2, 2)	0.0000000E+00	0.0000000E+00
E(2, 2, 3)	1.000000	0.0000000E+00
E(2, 2, 4)	0.0000000E+00	0.0000000E+00
E(2, 3, 1)	0.0000000E+00	0.0000000E+00
E(2, 3, 2)	0.0000000E+00	0.0000000E+00
E(2, 3, 3)	0.0000000E+00	0.0000000E+00
E(2, 3, 4)	0.0000000E+00	0.0000000E+00
E(3, 1, 1)	1.000000	0.0000000E+00
E(3, 1, 2)	0.0000000E+00	0.0000000E+00
E(3, 1, 3)	0.0000000E+00	0.0000000E+00
E(3, 1, 4)	1.000000	0.0000000E+00
E(3, 2, 1)	0.0000000E+00	0.0000000E+00
E(3, 2, 2)	1.000000	0.0000000E+00
E(3, 2, 3)	0.0000000E+00	0.0000000E+00
E(3, 2, 4)	1.000000	0.0000000E+00
E(3, 3, 1)	0.0000000E+00	0.0000000E+00
E(3, 3, 2)	0.0000000E+00	0.0000000E+00
E(3, 3, 3)	0.0000000E+00	0.0000000E+00
E(3, 3, 4)	0.0000000E+00	0.0000000E+00
E(4, 1, 1)	1.000000	0.0000000E+00
E(4, 1, 2)	0.0000000E+00	0.0000000E+00

E(4, 1, 3)	0.0000000E+00	0.0000000E+00
E(4, 1, 4)	1.000000	0.0000000E+00
E(4, 2, 1)	1.000000	0.0000000E+00
E(4, 2, 2)	0.0000000E+00	0.0000000E+00
E(4, 2, 3)	1.000000	0.0000000E+00
E(4, 2, 4)	0.0000000E+00	0.0000000E+00
E(4, 3, 1)	0.0000000E+00	0.0000000E+00
E(4, 3, 2)	0.0000000E+00	0.0000000E+00
E(4, 3, 3)	0.0000000E+00	0.0000000E+00
E(4, 3, 4)	0.0000000E+00	0.0000000E+00
E(5, 1, 1)	0.0000000E+00	0.0000000E+00
E(5, 1, 2)	0.0000000E+00	0.0000000E+00
E(5, 1, 3)	1.000000	0.0000000E+00
E(5, 1, 4)	1.000000	0.0000000E+00
E(5, 2, 1)	1.000000	0.0000000E+00
E(5, 2, 2)	0.0000000E+00	0.0000000E+00
E(5, 2, 3)	0.0000000E+00	0.0000000E+00
E(5, 2, 4)	0.0000000E+00	0.0000000E+00
E(5, 3, 1)	0.0000000E+00	0.0000000E+00
E(5, 3, 2)	0.0000000E+00	0.0000000E+00
E(5, 3, 3)	0.0000000E+00	0.0000000E+00
E(5, 3, 4)	0.0000000E+00	0.0000000E+00
NM(1, 1)	2.000000	0.0000000E+00
NM(1, 2)	2.000000	0.0000000E+00
NM(1, 3)	2.000000	0.0000000E+00
NM(2, 1)	2.000000	0.0000000E+00
NM(2, 2)	2.000000	0.0000000E+00
NM(2, 3)	10.00000	0.0000000E+00
NM(3, 1)	2.000000	0.0000000E+00
NM(3, 2)	2.000000	0.0000000E+00
NM(3, 3)	10.00000	0.0000000E+00
NM(4, 1)	2.000000	0.0000000E+00
NM(4, 2)	2.000000	0.0000000E+00

NM(4, 3)	10.00000	0.0000000E+00
NM(5, 1)	2.000000	0.0000000E+00
NM(5, 2)	1.000000	0.0000000E+00
NM(5, 3)	10.00000	0.0000000E+00
Y(1, 1, 1)	0.0000000E+00	0.0000000E+00
Y(1, 1, 2)	0.0000000E+00	0.0000000E+00
Y(1, 1, 3)	0.0000000E+00	0.0000000E+00
Y(1, 1, 4)	0.0000000E+00	0.0000000E+00
Y(1, 2, 1)	0.0000000E+00	0.0000000E+00
Y(1, 2, 2)	1.000000	0.0000000E+00
Y(1, 2, 3)	0.0000000E+00	0.0000000E+00
Y(1, 2, 4)	0.0000000E+00	0.0000000E+00
Y(1, 3, 1)	0.0000000E+00	0.0000000E+00
Y(1, 3, 2)	0.0000000E+00	0.0000000E+00
Y(1, 3, 3)	0.0000000E+00	0.0000000E+00
Y(1, 3, 4)	0.0000000E+00	0.0000000E+00
Y(2, 1, 1)	0.0000000E+00	0.0000000E+00
Y(2, 1, 2)	0.0000000E+00	0.0000000E+00
Y(2, 1, 3)	0.0000000E+00	0.0000000E+00
Y(2, 1, 4)	0.0000000E+00	0.0000000E+00
Y(2, 2, 1)	0.0000000E+00	0.0000000E+00
Y(2, 2, 2)	0.0000000E+00	0.0000000E+00
Y(2, 2, 3)	1.000000	0.0000000E+00
Y(2, 2, 4)	0.0000000E+00	0.0000000E+00
Y(2, 3, 1)	0.0000000E+00	0.0000000E+00
Y(2, 3, 2)	0.0000000E+00	0.0000000E+00
Y(2, 3, 3)	0.0000000E+00	0.0000000E+00
Y(2, 3, 4)	0.0000000E+00	0.0000000E+00
Y(3, 1, 1)	0.0000000E+00	0.0000000E+00
Y(3, 1, 2)	0.0000000E+00	0.0000000E+00
Y(3, 1, 3)	0.0000000E+00	0.0000000E+00
Y(3, 1, 4)	0.0000000E+00	0.0000000E+00
Y(3, 2, 1)	0.0000000E+00	0.0000000E+00

Y(3, 2, 2)	1.000000	0.0000000E+00
Y(3, 2, 3)	0.0000000E+00	0.0000000E+00
Y(3, 2, 4)	0.0000000E+00	0.0000000E+00
Y(3, 3, 1)	0.0000000E+00	0.0000000E+00
Y(3, 3, 2)	0.0000000E+00	0.0000000E+00
Y(3, 3, 3)	0.0000000E+00	0.0000000E+00
Y(3, 3, 4)	0.0000000E+00	0.0000000E+00
Y(4, 1, 1)	0.0000000E+00	0.0000000E+00
Y(4, 1, 2)	0.0000000E+00	0.0000000E+00
Y(4, 1, 3)	0.0000000E+00	0.0000000E+00
Y(4, 1, 4)	0.0000000E+00	0.0000000E+00
Y(4, 2, 1)	0.0000000E+00	0.0000000E+00
Y(4, 2, 2)	0.0000000E+00	0.0000000E+00
Y(4, 2, 3)	1.000000	0.0000000E+00
Y(4, 2, 4)	0.0000000E+00	0.0000000E+00
Y(4, 3, 1)	0.0000000E+00	0.0000000E+00
Y(4, 3, 2)	0.0000000E+00	0.0000000E+00
Y(4, 3, 3)	0.0000000E+00	0.0000000E+00
Y(4, 3, 4)	0.0000000E+00	0.0000000E+00
Y(5, 1, 1)	0.0000000E+00	0.0000000E+00
Y(5, 1, 2)	0.0000000E+00	0.0000000E+00
Y(5, 1, 3)	0.0000000E+00	0.0000000E+00
Y(5, 1, 4)	0.0000000E+00	0.0000000E+00
Y(5, 2, 1)	0.0000000E+00	0.0000000E+00
Y(5, 2, 2)	0.0000000E+00	0.0000000E+00
Y(5, 2, 3)	1.000000	0.0000000E+00
Y(5, 2, 4)	0.0000000E+00	0.0000000E+00
Y(5, 3, 1)	0.0000000E+00	0.0000000E+00
Y(5, 3, 2)	0.0000000E+00	0.0000000E+00
Y(5, 3, 3)	0.0000000E+00	0.0000000E+00
Y(5, 3, 4)	0.0000000E+00	0.0000000E+00
X(1, 1)	0.0000000E+00	0.9644433
X(1, 2)	0.0000000E+00	0.0000000E+00

X(1, 3)	1.000000	0.0000000E+00
X(1, 4)	0.0000000E+00	0.0000000E+00
X(2, 1)	0.0000000E+00	0.0000000E+00
X(2, 2)	1.000000	0.0000000E+00
X(2, 3)	0.0000000E+00	0.0000000E+00
X(2, 4)	0.0000000E+00	0.0000000E+00
X(3, 1)	0.0000000E+00	0.0000000E+00
X(3, 2)	0.0000000E+00	0.0000000E+00
X(3, 3)	1.000000	0.0000000E+00
X(3, 4)	0.0000000E+00	0.0000000E+00
X(4, 1)	0.0000000E+00	0.0000000E+00
X(4, 2)	1.000000	0.0000000E+00
X(4, 3)	0.0000000E+00	0.0000000E+00
X(4, 4)	0.0000000E+00	0.0000000E+00

ROW	SLACK OR SURPLUS	DUAL PRICE
1	3.322330	1.639554
2	2.000000	0.0000000E+00
3	0.0000000E+00	0.0000000E+00
4	0.0000000E+00	0.0000000E+00
5	2.000000	0.0000000E+00
6	0.0000000E+00	0.0000000E+00
7	0.0000000E+00	0.0000000E+00
8	0.0000000E+00	0.0000000E+00
9	0.0000000E+00	0.0000000E+00
10	0.0000000E+00	0.0000000E+00
11	0.0000000E+00	0.0000000E+00
12	0.0000000E+00	0.0000000E+00
13	0.0000000E+00	0.0000000E+00
14	0.0000000E+00	0.0000000E+00

APPENDIX 5

COMPUTER PROGRAM FOR THE SIMULATED ANNEALING ALGORITHM PRESENTED IN SECTION 5.2 USING DATA FROM PROBLEM 1 PRESENTED IN SECTION 5.5

```
C*****************************************************************************************
C*                    LIST OF THE VARIABLES USED IN THIS PROGRAM                         *
C*****************************************************************************************
C     BTP       BEST TOTAL PRODUCTIVITY FOUND SO FAR
C     BX        BEST SOLUTION FOUND
C     C         NUMBER OF CELLS
C     D         DEMAND (NUMBER OF UNITS REQUIRED)
C     EMC       INTER CELL COST
C     IMC       INTRA CELL COST
C     M         THE NUMBER OF MACHINES IN THE PROBLEM
C     MMAX      MAXIMUM NUMBER OF MACHINES IN EACH CELL
C     NIRAT     NUMBER OF ITERATION AT EACH TEMPERATURE LEVEL
C     NTP       NEIGHBOR SOLUTION (NEW TOTAL PRODUCTIVITY)
C     NPP       NUMBER OF PROCESS PLANS
C     R1        THE SELECTED MACHINE TO MOVE (RANDOMLY)
C     R2        THE SELECTED CELL  (RANDOMLY)
C     O         OUTPUT
C     P         NUMBER OF PARTS IN THE PROBLEM
C     S         SELLING PRICE
C     SUMI      TOTAL NUMBER OF ENTER CELL MOVEMENT
C     SUMI      TOTAL NUMBER OF INTRA CELL MOVEMENT
C     SUMT      TOTAL MATERIAL HANDLING COST
C     T         CURRENT TEMPERATURE  (CURRENT SOLUTION)
C     TP        CURRENT TOTAL PRODUCTIVITY
C     TSTOP     STOPPING TEMPERATURE
C     TFAC      DECREMANTAL FACTOR
```

```
C*************************************************************************************
C*                              MIN PROGRAM                                          *
C*************************************************************************************
    INTEGER Y(150,150),X(150,150),D(150),S(150),NX(150,150),MC(150),
    SUMI(130),SUME(130), SUM1,SUM2,SUMT,SUM(130),PR(130),LSUM(130),
    BX(130,130)
    REAL  NTP,T
    N1=586754699
    N2=326574579
    N3=865473439
C  INPUTS
    MAXM=5
    EMC=4.6
    IMC=2
    T=10
    NIRAT=64
    TFAC=0.99
    TSTOP=0.05

C*************************************************************************************
C*                     OPEN, READ AND WRITE INPUT FILE                               *
C*************************************************************************************
    OPEN (UNIT=20,FILE='LP1.FOR')
    OPEN (UNIT=10,STATUS='OLD',FILE='ABD.TXT')
    READ(20,*)P,M,C
    WRITE(10,*)P,M,C
    DO 10 I=1,P
      READ(20,*)NPP(I)
10  CONTINUE
     DO 20 I=1,P
     WRITE(10,*)NPP(I)
     WRITE(10,*)'NPP(I)=',NPP(I)
```

```
20  CONTINUE
DO 30 I=1,P
DO 40 R=1,NPP(I)
     READ(20,*)(Y(I,R,J),J=1,M)
40  CONTINUE
30  CONTINUE
     DO 50 I=1,P
        DO 60 R=1,NPP(I)
               WRITE (10,*)(Y(I,R,J),J=1,M)
60  CONTINUE
50  CONTINUE
     DO 70 K=1,C
     READ (20,*)(X(K,J),J=1,M)
70  CONTINUE
     DO 80 K=1,C
     WRITE (10,*)(X(K,J),J=1,M)
80 CONTINUE
     READ (20,*)(D(I),I=1,P)
     WRITE(10,*)(D(I),I=1,P)
     READ (20,*)(S(I),I=1,P)
     WRITE (10,*)(S(I),I=1,P)

C**************************************************************************************
C*                    CALCULATE THE INITIAL SOLUTION                                 *
C**************************************************************************************
     OUT=0
     DO 85 I=1,P
        OUT=OUT+D(I)*S(I)
        WRITE(10,*)'OUT=',OUT
85  CONTINUE
     WRITE(*,*)'OUTPUT=',OUT
     SUMT=0
     DO 90  I=1,P
```

```
      DO 95 R=1,NPP(I)
            SUM1=0
            SUM2=0
            DO 100  K=1,C
      NSW=0
      DO 105 J=1,M
                              IF((Y(I,R,J).EQ.1).AND.(X(K,J).EQ.1))THEN
                                    SUM1=SUM1+1
                                    NSW=1
                                    END IF
105               CONTINUE
   IF(NSW.EQ.1)THEN
      SUM2=SUM2+1
   END IF
100 CONTINUE
   SUM(R)=SUM1*D(I)*IMC+(SUM2-1)*D(I)*EMC
   WRITE(10,*)'SUM(R)=',SUM(R)
   WRITE(10,*)'SUM(',R,')=',SUM(R)
95  CONTINUE
   LSUM(I)=SUM(1)
   Z(I)=1
   DO 110 R=2,NPP(I)
      IF(SUM(R).LT.LSUM(I))THEN
            LSUM(I)=SUM(R)
            Z(I)=R
      ENDIF
110 CONTINUE
   WRITE(10,*)'SUM(',R,')=',SUM(R)
   WRITE(10,*)'Z(',I,')=',Z(I)
   SUMT=SUMT+LSUM(I)
   WRITE(10,*)'SUMT=',SUMT
90 CONTINUE
   TP=OUT/SUMT
```

```
      WRITE(10,*)'TP=',TP
      BTP=TP

C***************************************************************************************
C*                            GENERATE A NEW SOLUTION                                  *
C***************************************************************************************
      DO 115 K=1,C
         DO 53 J=1,M
               IF (X(K,J).EQ.1) MC(J)=K
         NX(K,J)=X(K,J)
         BX(K,J)=X(K,J)
53    CONTINUE
115   CONTINUE
      DO 13  I=1,P
         NZ(I)=Z(I)
         BZ(I)=Z(I)
13    CONTINUE
205   IE=0
120   IE=IE+1

C***************************************************************************************
C*        THE LINES FROM STATEMENT LABEL 130 TO LABEL 150 ARE ADAPTED                  *
C*                            FROM COTRUVO(1993)                                       *
C***************************************************************************************
130   CALL RAND1(N1,RD1)
      R1 =INT(M*RD1 + 1)
      WRITE(10,*)'THE MACHINE TO MOVE =',R1
      CALL RAND2(N2,RD2)
      R2 =INT(C*RD2 + 1)
      WRITE(10,*)'THE CELL TO MOVE THE MACHINE TO  =',R2
      IF (MC(R1).EQ.R2) GOTO 130
      SE=0
      DO 140 J=1,M
```

```
      SE = SE + NX(R2,J)
140 CONTINUE
      IF (SE.GE.MAXM) GOTO 130
      NX(MC(R1),R1) =0
      NX(R2,R1) = 1
      MC(R1)=R2
      DO 150 K=1,C
      WRITE(10,*)(NX(K,J),J=1,M)
150 CONTINUE
      SUMT=0
      DO 160  I=1,P
         DO 170 R=1,NPP(I)
                SUM1=0
                SUM2=0
         DO 180  K=1,C
                              NSW=0
                              DO 190 J=1,M
                                    IF((Y(I,R,J).EQ.1).AND.(NX(K,J).EQ.1))THEN
                                    SUM1=SUM1+1
                                    NSW=1
                                    END IF
190                       CONTINUE
                              IF(NSW.EQ.1)THEN
                              SUM2=SUM2+1
                              END IF
180                 CONTINUE
         SUM(R)=SUM1*D(I)*IMC+(SUM2-1)*D(I)*EMC
         WRITE(10,*)'SUM(R)=',SUM(R)
         WRITE(10,*)'SUM(',R,')=',SUM(R)
170           CONTINUE
                LSUM(I)=SUM(1)
                NZ(I)=1
                DO 200 R=2,NPP(I)
```

```
                IF(SUM(R).LT.LSUM(I))THEN
                LSUM(I)=SUM(R)
                NZ(I)=R
                ENDIF
200   CONTINUE
         WRITE(10,*)'SUM(',R,')=',SUM(R)
         WRITE(10,*)'NZ(',I,')=',NZ(I)
         SUMT=SUMT+LSUM(I)
         WRITE(10,*)'SUMT=',SUMT
160 CONTINUE
      NTP=OUT/SUMT
      WRITE(10,*)'NTP=',NTP
      IF(NTP.GE.TP)THEN
         TP=NTP
         NX(MC(R1),R1)=0
         NX(R2,R1)=1
         MC(R1)=R2
      ENDIF
      IF(NTP.GE.BTP)THEN
         BTP=NTP
         NX(MC(R1),R1)=0
         NX(R2,R1)=1
         MC(R1)=R2
         DO 210 I=1,P
               BZ(I)=NZ(I)
210   CONTINUE
      DO 220 K=1,C
         DO 230 J=1,M
            BX(K,J)=NX(K,J)
230 CONTINUE
220 CONTINUE
      ENDIF
      IF(NTP.LT.TP)THEN
```

```
      DZ=ABS(NTP-TP)
      ACC=EXP(-DZ/T)
      WRITE(10,*)'ACC=',ACC
      CALL RAND3(N3,RD3)
      WRITE(10,*)'RD3=',RD3
      IF(RD3.LE.ACC)THEN
         TP=NTP
         NX(MC(R1),R1)=0
         NX(R2,R1)=1
         MC(R1)=R2
      ELSE
         NTP=TP
         NX(MC(R1),R1)=0
         NX(R2,R1)=1
      ENDIF
      ENDIF
      IF(IE.LT.NIRAT)GOTO 120
     T=T*TFAC
      IF(T.LE.TSTOP)THEN
         GOTO 230
      ELSE
         GOTO 205
      ENDIF
230   DO 240 K=1,C
      WRITE(10,*)(BX(K,J),J=1,M)
240   CONTINUE
      WRITE (*,*)'PRODUCTIVITY=',BTP
      WRITE (10,*)
      WRITE (10,245)
      DO 250 K=1,C
         WRITE (10,255)K
         DO 260 J=1,M
            IF(BX(K,J).EQ.1)WRITE(10,265)J
```

```
260    CONTINUE
250 CONTINUE
245 FORMAT(2X,'CELL',4X,'MACHINE')
255 FORMAT(2X,I2)
265 FORMAT (10X,I2)

C*******************************************************************************
C*                              PART ASSINMENT                                 *
C*******************************************************************************
     DO 270 I=1,P
        DO 280 K=1,C
           SUM(K)=0
           SUM1=0
           DO 290 J=1,M
     IF ((Y(I,BZ(I),J).EQ.1).AND.(BX(K,J).EQ.1)) THEN
        SUM1=SUM1+1
     END IF
290    CONTINUE
           SUM(K)=SUM1
280 CONTINUE
     LSUM(I)=SUM(1)
     PR(I)=1
        DO 285 K=2,C
           IF (SUM(K).GT.LSUM(I))THEN
           LSUM(I)=SUM(K)
           PR(I)=K
           END IF
285    CONTINUE
270 CONTINUE
     WRITE (10,295)
     DO 300 K=1,C
     WRITE(10,305)K
        DO 310 I=1,P
```

```
        IF(PR(I).EQ.K)WRITE(10,315)I
310 CONTINUE
300 CONTINUE
295 FORMAT(2X,'CELL',4X,'PART')
305 FORMAT(1X,I2)
315 FORMAT(8X,I3)
    WRITE (10,325)
    DO 320 I=1,P
       WRITE(10,335)I
          DO 330 R=1,NPP(I)
    WRITE(10,305)I
       IF(BZ(I).EQ.R)WRITE(10,345)R
330 CONTINUE
320 CONTINUE
325 FORMAT(2X,'PART',4X,'PROC. PLANE')
335 FORMAT(1X,I2)
345 FORMAT(8X,I3)
    END

C***************************************************************************************
C*          THE FOLLOWING SUBROUTINES ARE ADAPTED FROM ARE ADAPTED                   *
C*                              FROM COTRUVO(1993)                                   *
C***************************************************************************************
    SUBROUTINE RAND1(N1,RD1)
    L=316227
    N1=N1*L
    RN=N1
    RD1=RN/(2.**31 - 1)
    RD1=ABS(RD1)
    RETURN
    END
    SUBROUTINE RAND2(N2,RD2)
    L=316227
```

```
N2=N2*L
RN=N2
RD2=RN/(2.**31 - 1)
RD2=ABS(RD2)
RETURN
END
SUBROUTINE RAND3(N3,RD3)
L=316227
N3 =N3*L
RN=N3
RD3 =RN/(2.**31 - 1)
RD3 = ABS(RD3)
RETURN
END
```

VITA AUCTORIS

NAME:	Abduelghani Abduelmula
PLACE OF BIRTH	Zawya, Libya
YEAR OF BIRTH	1956
EDUCATION	Al-fateh University, Tripoli, Libya
	1982-1986 B. Sc.
	Al-fateh University, Tripoli, Libya
	1987-1992 M. Sc.
	University of Windsor, Windsor, Ontario, Canada
	1995-2000 Ph.D.